A security protocol for authentication of binding updates in Mobile IPv6

Andrew Georgiades

June 2011

A thesis submitted to Middlesex University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Engineering and Information Sciences

Hendon. NW4 4BT

United Kingdom

I Abstract

Wireless communication technologies have come along way, improving with every generational leap. As communications evolve so do the system architectures, models and paradigms. Improvements have been seen in the jump from 2G to 3G networks in terms of security. Yet these issues persist and will continue to plague mobile communications into the leap towards 4G networks if not addressed.

4G will be based on the transmission of Internet packets only, using an architecture known as mobile IP. This will feature many advantages, however security is still a fundamental issue to be resolved. One particular security issue involves the route optimisation technique, which deals with binding updates. This allows the corresponding node to by-pass the home agent router to communicate directly with the mobile node. There are a variety of security vulnerabilities with binding updates, which include the interception of data packets, which would allow an attacker to eavesdrop on its contents, breaching the users confidentiality, or to modify transmitted packets for the attackers own malicious purposes. Other possible vulnerabilities with mobile IP include address spoofing, redirection and denial of service attacks. For many of these attacks, all the attacker needs to know is the IPv6 addresses of the mobile's home agent and the corresponding node.

There are a variety of security solutions to prevent these attacks from occurring. Two of the main solutions are cryptography and authentication. Cryptography allows the transmitted data to be scrambled in an undecipherable way resulting in any intercepted packets being illegible to the attacker. Only the party possessing the relevant key will be able to decrypt the message. Authentication is the process of verifying the identity of the user or device one is in communication with. Different authentication architectures exist however many of them rely on a central server to verify the users, resulting in a possible single point of attack. Decentralised authentication mechanisms would be more appropriate for the nature of mobile IP and several protocols are discussed. However they all posses' flaws, whether they be overly resource intensive or give away vital address data, which can be used to mount an attack. As a result location privacy is investigated in a possible attempt at hiding this sensitive data. Finally, a security solution is proposed to address the security vulnerabilities found in binding updates and attempts to overcome the weaknesses of the examined security solutions.

The security protocol proposed in this research involves three new security techniques. The first is a combined solution using Cryptographically Generated Addresses and Return Routability, which are already established solutions, and then introduces a new authentication procedure, to create the Distributed Authentication Protocol to aid with privacy, integrity and authentication. The second is an enhancement to Return Routability called Dual Identity Return Routability, which provides location verification authentication for multiple identities on the same device. The third security technique is called Mobile Home Agents, which provides device and user authentication while introducing location privacy and optimised communication routing. All three security techniques can be used together or individually and each needs to be passed before the binding update is accepted.

Cryptographically Generated Addresses asserts the users ownership of the IPv6 address by generating the interface identifier by computing a cryptographic one-way hash function from the users' public key and auxiliary parameters. The binding between the public key and the address can be verified by re-computing the hash value and by comparing the hash with the interface identifier. This method proves ownership of the address, however it does not prove the address is reachable.

After establishing address ownership, Return Routability would then send two security tokens to the mobile node, one directly and one via the home agent. The mobile node would then combine them together to create an encryption key called the binding key allowing the binding update to be sent securely to the correspondent node. This technique provides a validation to the mobile nodes' location and proves its ownership of the home agent.

Return Routability provides a test to verify that the node is reachable. It does not verify that the IPv6 address is owned by the user. This method is combined with Cryptographically Generated Addresses to provide best of both worlds.

The third aspect of the first security solution introduces a decentralised authentication mechanism. The correspondent requests the authentication data from both the mobile node and home agent. The mobile sends the data in plain text, which could be encrypted with the binding key and the home agent sends a hash of the data. The correspondent then converts the data so both are hashes and compares them. If they are the same, authentication is successful. This provides device and user authentication which when combined with Cryptographically Generated Addresses and Return Routability create a robust security solution called the Distributed Authentication Protocol.

The second new technique was designed to provide an enhancement to a current security solution. Dual Identity Return Routability builds on the concept of Return Routability by providing two Mobile IPv6 addresses on a mobile device, giving the user two separate identities. After establishing address ownership with Cryptographically Generated Addresses, Dual Identity Return Routability would then send security data to both identities, each on a separate network and each having heir own home agents, and the mobile node would then combine them together to create the binding key allowing the binding update to be sent securely to the correspondent node. This technique provides protection against address spoofing as an attacker needs two separate ip addresses, which are linked together. Spoofing only a single address will not pass this security solution.

One drawback of the security techniques described, however, is that none of them provide location privacy to hide the users IP address from attackers. An attacker cannot mount a direct attack if the user is invisible.

The third new security solution designed is Mobile Home Agents. These are software agents, which provide location privacy to the mobile node by acting as a proxy between it and the network. The

Mobile Home Agent resides on the point of attachment and migrates to a new point of attachment at the same time as the mobile node. This provides reduced latency communication and a secure environment for the mobile node.

These solutions can be used separately or combined together to form a super security solution, which is demonstrated in this thesis and attempts to provide proof of address ownership, reachability, user and device authentication, location privacy and reduction in communication latency. All these security features are design to protect against one the most devastating attacks in Mobile IPv6, the false binding update, which can allow an attacker to impersonate and deny service to the mobile node by redirecting all data packets to itself.

The solutions are all simulated with different scenarios and network configurations and with a variety of attacks, which attempt to send a false binding update to the correspondent node. The results were then collected and analysed to provide conclusive proof that the proposed solutions are effective and robust in protecting against the false binding updates creating a safe and secure network for all.

II Acknowledgements

I would like to thank my supervisors for helping me and putting up with me. Dr Yuan Luo, thank you for taking me on as your student and guiding me to this point. Dr. Aboubaker Lasebae, thank you for your support and guidance throughout the course of this project. Prof. Richard Comley, I have benefited from your vast experience thank you for your time.

Special thank you to Prof. Colin Tully, your understanding has given my work a new lease of life. Special thanks to the research administration staff at Middlesex University especially, Emma Warne, Kerry Lane , Dr. Robert Pleas and Dr. Claudia Kalay.

Thank you to Dr Luminita Vasiu for helping me to begin my journey. And of course thanks to everyone in CCM for your support.

During the course of the PhD I have interacted with other PhD students who have had an influence on my work. Special thank you to Saurabh Bansal for your knowledge of Mobile IP and for your guidance and insights. Thank you to Michael Cheng for your knowledge of elliptic curve cryptography. Thank you, Agozie Eneh for you knowledge of authentication.

Thanks to Mr. Gerald Osei-kofi for your help with technical matters. Special thanks to all the PhD candidates I haven't mentioned for making my experience at Middlesex a pleasant one and for creating a friendly environment for research. Birinder Sandhawalia, Abhishek Agrawal, Rachid Bencheikh, Nikolaos Papamichail, Cristian Donciulescu, Saif ur Rehman, Fatima Sheikh, John Salisbury, Stephen Batty, Kunbin Hong, Mohammad Mostafa Kamal, Manohar Menon, Evangelos Moustakas, Dili Ojukwu, Anja Schanzenberger, Vooi Voon Yap, Keh Kok Yong.

And everyone else I have met in the duration of the course. I also thank anyone who has helped me directly or indirectly to complete this project.

Special thank you to my parents and sister for your love, support and sacrifice.

Thanks to my friends (Grant, Mo, Ahmed, Steve, Roger) for keeping me sane and entertained.

And very warm thank you to Manju Rani who lit the way when I couldn't see the path.

And of course thank you Andrew Georgiades (yes myself!) for doing the work. Well Done!

III Table of Contents

IV List of Figures

V List of Tables

VI List of Notations

MN	A mobile node
CN	A correspondent node
MN ⟶ CN	Mobile node sends a message to correspondent
HA	Home Agent
Hash(m)	A hash of message m
K^+	Public Key
K^-	Private Key
MNK^+	Mobile nodes public key
A ⟶ B	Node A sends a message to B
A ⟶ B(HoA)	Node A sends a message to B at its home address
A ⟶ B(CoA)	Node A sends a message to B at its care-of address
HoA	Mobile node's home address
CoA	Mobile node's care-of address
$MAC_K(m)$	Message authentication code computed on message m with key K
$K_{bm}(BU)$	Binding update encrypted with the binding key.
Ka	Secret value derived from a hash function.
Ni	Secret value added as a component of a hash function.
H_1	The first hash in a chain of hash values.
H_N	Total number of hash values in a chain.
CGA	Cryptographically Generated Address
RR	Return Routability
DIRR	Dual Identity Return Routability
DAP	Distributed Authentication Protocol
MHA	Mobile Home Agent.
Cga1	CGA impact with attacker using its own CGA address.
Cga2	CGA impact with attacker attempting to spoof the MNs Home Agent
RR1	RR with attacker using HA as HoT .
RR2	RR with attacker using it's self to spoof the HA
ATK	Attacker
MNH - CNH	Mobile Node at home to Correspondent Node at home
MNA - CNH	Mobile Node at Away from home to Correspondent Node at home
MNH - CNA	Mobile Node at home to Correspondent Node away
MNA - CNA	Mobile Node at Away from home to Correspondent Node away
N_H	Nonce for the Home Token
N_C	Nonce for the Care-of Token
N_{H2}	Nonce for the second Home Token
N_{C2}	Nonce for the second Care-of Token
K_{cn}	Public key for the Correspondent Node.

VII List of Publications

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. “Location Privacy in Mobile IPv6 Distributed Authentication Protocol Using Mobile Home Agents”, Recent Advances in Electronics, Hardware, Wireless and Optical Communications. Proceedings of the 8th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications (EHAC ’09), Cambridge, UK, February 21-23, 2009. WSEAS Press, ISBN: 978-960-474-053-6

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. “Introducing Mobile Home Agents into the Distributed Authentication Protocol to Achieve Location Privacy in Mobile IPv6” INTERNATIONAL JOURNAL of COMMUNICATIONS, (Issue 3, Volume 2, 2008) http://www.naun.org/journals/communications/2008.htm

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. "Dual Identity Return Routability for the Security of Mobile Ipv6 Binding Updates within the Distributed Authentication Protocol". WSEAS Proceedings in the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Agios Nikolaos, Crete Island, Greece (August 2006)

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. “Distributed Authentication Protocol *Utilizing* Dual Identity Return Routability for the Security of Binding Updates within Mobile Ipv6". WSEAS Transactions on Communications, August 2006.

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. "*Distributed Authentication Protocol for the security of Binding Updates in Mobile IPv6*". WSEAS Proceedings in the 9th WSEAS International Conference on Communications, Vouliagmeni, Athens, Greece (July 2005) ISBN:960-8457-29-7

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. "*Binding Update Security for Mobile Ipv6 using the Distributed Authentication Protocol*". WSEAS Transactions on Communications, Issue 9, volume 4, September 2005, ISSN 1109-2742

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. "*Trinity Protocol for Authentication of Binding Updates in Mobile IPv6*", 3rd WSEAS International Conference on Information Security, Copacabana, Rio de Janeiro, Brazil, October 12-15, 2004.

- **A. Georgiades**, Y. Luo, A. Lasebae, R. Comley. "*Trinity Protocol for Authentication of Binding Updates in Mobile IPv6*", WSEAS Transactions on Communications, Issue 3, volume 3, July 2004, ISSN 1109-2742.

"**The journey of a thousand miles begins with a single step**"

-Confucius

500 B.C

"…on the accelerator!"

-Andrew Georgiades

2004 A.D

Chapter 1: Introduction

1.1 Introduction

Networking has always been vulnerable to a variety of attacks and the next generation of mobile communications is no different. 4G will be based on the transmission of Internet packets only, using an architecture known as mobile IP. This will feature many advantages however security is still a fundamental issue to be resolved. One particular security issue involves the route optimisation technique, which deals with binding updates [1]. This allows the corresponding node to by-pass the home agent router and communicate directly with the mobile node. Attempts have been made to resolve this issue with the introduction of Return Routability and Cryptographically Generated Address which is discussed in chapter 4.

The home agent has a static address while the mobile node's address changes every time it moves to a new location with a new point of attachment. The home agent keeps track of the mobile node's current address, so that if a correspondent does not know it, it may send the packets to the home address, which will forward the packets to the mobile node [2]. By bypassing the home address with the binding update route optimisation option, the speed of the delivery of packets increases.

Chapter two will investigate the mobile IP architecture in more depth and also look at some of the surrounding protocols. There are a variety of security vulnerabilities with binding updates [3], which will be explored in this thesis including the interception of data packets [4], which would allow an attacker to eavesdrop on its contents breaching the users confidentiality or to modify transmitted packets for the attackers own malicious purposes. Other possible vulnerabilities with mobile IP include address spoofing [5], redirection and denial of service attacks [6]. For many of these attacks all the attacker needs to know is the IPv6 addresses of the mobile's home agent and the corresponding node. By looking at each of these vulnerabilities, a detailed picture can be constructed of the weaknesses in the current mobile IP architecture leading to an understanding of which security solutions need to be applied and where. Also, many of the different attacks may be possible because of a single or common vulnerability and this must be addressed.

To solve the vulnerability issues of binding updates, a variety of security threats and solutions will be investigated in an attempt to create a unique security solution with the advantages of the previous security solutions yet without any of their disadvantages or drawbacks. Numerous security solutions have been proposed and in Chapter Four each will be investigated and have their advantages and disadvantages explored.

The two main types of security are encryption and authentication.

Encryption protects the confidentiality of the data and comes in two flavours, symmetric and asymmetric [4]. The former is useful for low powered devices and participants use the same key to encrypt and decrypt. The problem is how to distribute the key without it being intercepted. Asymmetric keys are split into encryption and decryption keys. This is useful for the distribution of the keys and can help with authentication with the use of digital signatures [4]. The drawback however is that processing consumption is 100 – 1000 times that of symmetric cryptography. This can be reduced somewhat with the implementation of elliptic curve cryptography [7] which is a lightweight public key cryptographic solution.

Authentication allows users to verify that they are communicating with validated participants. Different authentication systems will be explored, such as Kerberos [8] that perform authentication by referring to a central authentication database to compare users credentials. An understanding of the main components of different authentication systems will allow for commonalities to be found which will be applied to the final proposed solution. These components will include techniques such as hashes [9], digital signatures [4], address based keys [10] and cryptographically generated addresses [11].

More elaborate systems such as IPSEC [12] and RADIUS [13] based on AAA Authentication, Authorization and Accounting [14], will also be investigated to determine if the implementation of a security architecture is necessary and if the utilization of a central authentication authority is required. It will also be determined if the cost in resources to utilize these architectures is beyond what it realistically expected from a mobile device, effectively reducing the users quality of service.

Security protocols, which have been specifically designed for the protection of binding updates such as, Bake/2 [15] and CAM [16], will also be covered in the fourth chapter. Their main purpose is to protect binding updates from eavesdropping, modification and DOS attacks. Their protocols will be examined to determine any flaws in their operation so as not to repeat the same fundamental errors in the proposed security solution. The security solution is proposed in chapter five, which includes a detailed explanation of the architecture, the messages communicated and the different types of security to be implemented.

1.2 Problem Definition

The binding update route optimisation protocol suffers from security vulnerabilities, which allow attackers to send false binding updates to redirect data traffic for interception and eavesdropping of packets or the prevention of communication via denial of service attacks.

These problems exist because current security protocols don't effectively authenticate the legitimacy of the users or hide the location data of the home agent and care of address, leaving the participants vulnerable to attack. Many of the solutions that currently exist are resource intensive in terms of processing power required, which is unavailable with mobile devices.

The proposed solution will be composed of several security components. The first will attempt to address these issues by using existing security technologies without modifying the established mobile IPv6 architecture. Its main function is to aid in user and device authentication while providing an option for distributed authentication to aid with processor intensive situations. These combined features will protect the nodes from false binding updates attempting to hijack the session. However, the other more advanced solutions may have to modify the Mobile IPv6 architecture to meet their objectives.

1.3 Research Questions

Can attacks against the security vulnerabilities of the binding update route optimisation protocol, in Mobile IPv6, be prevented by developing a security solution using the existing infrastructure? The solution would have to solve the security issues of attacks, which send false binding updates to redirect data traffic for interception and eavesdropping of packets or the prevention of communication via denial of service attacks. And can the exchange of security keys be effectively facilitated without a central authentication authority perhaps by using a distributed solution.

Security solutions exist but contain flaws that attackers can take advantage of. Can an existing security solution or protocol be enhanced to improve its effectiveness and make it less vulnerable to certain types of attack? This would be done by researching the advantages and disadvantages of current security systems, such as Return Routability, and proposing modifications to enhance their effectiveness.

Triangle routing still takes place before route optimisation occurs and can incur significant latency issues. Can an enhancement to the mobile IPv6 infrastructure be introduced to reduce the latency of triangle routing packets to the Home Agent while maintaining or enhancing binding update security and location privacy? Would the introduction of Mobile Agent technology be able to contribute to the security of the network?

1.4 Limitations of Scope

There are numerous security attacks that can be mounted on any nodes in a Mobile IPv6 network. However the scope of this research will be specifically aiming to secure the binding update message to ensure the robustness of the route optimisation technique.

1.5 Main Contributions

The thesis provides three unique contributions:

A. **The Distributed Authentication Protocol.** This provides a de-centralised authentication solution within the existing Mobile IPv6 infrastructure, which can be used on its own or in combination with other security techniques.
B. **Dual Identity Return Routability.** This is an enhancement to Return Routability, which provides location authentication by using a second identity on an alternative network to transport half the security tokens needed to create a binding key.
C. **Mobile Home Agents.** This is a software agent, which emulates the functions of the Home Agent and operates from the point of attachment of which the Mobile Node is located. When the Mobile Node moves to another point of attachment, so does the Mobile Home Agent. This provides a layer of security to the Mobile Node, as the Mobile Home Agent can't be as easily attacked, due to its mobility and dynamic nature, unlike that of a static Home Agent. This allows for binding updates to be less susceptible to types of Denial of Service attack. It also provides a reduction in communication latency for packets destined for the Home Agent, as those packets are no longer routed to the home network first.

1.6 Originality of Intended Work

Binding updates are a very important optimisation within Mobile IPv6, which allow for faster packet transmission. Various security solutions have been proposed to protect these updates from vulnerabilities, which an attacker could take advantage of, causing a disruption to the normal operation of the network or breaching the confidentiality of a user's data. However the solutions that have been proposed suffer from drawbacks whether they be expensive in terms of resource consumption or by giving away fundamental location data, which the attacker may take advantage of. The proposed protocol attempts to be cost affective, prevent various attacks, provide location privacy and offers options for distributed authentication by spreading part of the processing to the home agent.

1.7 Research Methodology

- **Background research and literature review**

Research possible security threats and vulnerabilities associated with mobile IP binding updates. Then investigate current solutions that attempt to address these vulnerabilities and find the weaknesses in these solutions. The research will then continue by exploring the current types of security available in an attempt to find a more suitable solution.

- **Design proposed security solution.**

From the information gathered in the literature review, a solution will be put forward which addresses the needs of binding updates by eliminating the vulnerabilities and improving the security of previously proposed solutions.

- **Write and submit research papers.**

Using the information gathered in the literature review and the model of the proposed security solution, research papers will be created which will be submitted to international conferences.

- **Investigate simulation environment.**

Before simulations can be carried out, possible simulation environments must be investigated. This will consist of a combination of hardware and software requirements and the actual simulation software to be selected.

- **Create virtual security solution**

Create a virtual representation of the security solution within the simulation environment. The nodes will be modelled and the behaviour of the nodes in relation to each other and the protocol messages will be programmed. Security solution will be created using the open source Omnet network modelling software.

- **Test simulation**

The simulation will be tested to see if the protocol operates correctly. If so then further testing will be carried out with a variety of scenarios, which have been examined in the literature review, where possible attacks will take place to observe how the protocol reacts to them, and if the attacks are prevented or mitigated successfully.

- **Evaluation of results.**

The results of the simulations will be gathered and evaluated to see if any flaws are discovered and then possible improvements can be made to the design.

- **Thesis write-up**

1.8 Structure of Dissertation

The breakdown of the rest of this report is as follows:

Chapter 1 gives an introduction to the subject discusses the methodology to be employed.

Chapter 2 explores the background research and gives an overview of the of mobile IP technology and introduces the concept of the binding update route optimisation technique.

Chapter 3 discusses the flaws and possible attacks which can occur to binding updates.

Chapter 4 examines a variety of security solutions, which exist and may be used to remove the vulnerabilities of binding updates.

Chapter 5 introduces the proposed security solution and explains its components, operation and justification.

Chapter 6 Simulation and Results

Chapter 7 Analysis of the proposed solution.

Chapter 8 gives the conclusion of this report.

Chapter 2: Background Research

2.1 Generational Security Issues in Mobile Systems

Communication devices have become ubiquitous within society and with ever increasing features, such as m-commerce, video and data communications; the need for more sophisticated devices and infrastructure are needed to support them. However with ever increasing sophistication comes increased risks in security, and unfortunately people who will take advantage of them.

2.1.1 Security Issues in 2G

The first digital mobile communications service, 2G, was riddled with security flaws, ranging from denial of service attacks, impersonation, eavesdropping and hijacking of communications. The following gives a list of the security flaws of the architecture, which is described in [17].

1. Intentional jamming in a denial of service attacks
2. User de-registration request spoofing
3. Location update request spoofing
4. Camping on a false BS
5. Camping on a false BS/MS
6. Passive identity catching
7. Active identity catching
8. Impersonation of the network by suppressing encryption between the target user and the intruder
9. Impersonation of the network by suppressing encryption between the target user and the true network
10. Impersonation of the network by forcing the use of a compromised cipher key
11. Eavesdropping on user data by suppressing encryption between the target user and the intruder
12. Eavesdropping on user data by forcing the use of a compromised cipher key
13. Impersonation of the user through the use of by the network of a compromised authentication vector
14. Impersonation of the user through the use by the network of an eavesdropped authentication response
15. Hijacking outgoing calls in networks with encryption disabled
16. Hijacking outgoing calls in networks with encryption enabled
17. Hijacking incoming calls in networks with encryption disabled
18. Hijacking incoming calls in networks with encryption enabled

Fortunately as technology progressed most of the security weaknesses were solved. However some gaps in the security have persisted.

2.1.2 Security Issues in 3G

Some of the issues of 2G have been partially solved but some of them have not been solved at all. The partially solved issues imply that security solutions and models have been attempted but still require further work or don't cover the entire problem. These are the issues that should be addressed before the next generation leap takes place [17].

2.1.2.1 Unsolved Problems

1. Intentional jamming in denial of service attacks.
2. Camping on a false Base Station (BS)
3. Camping on a false Base Station (BS)/ Mobile Station (MS)

2.1.2.2 Partially Solved Problems

1. Impersonation of the network by forcing the use of a compromised cipher key
2. Hijacking outgoing calls in networks with encryption disabled
3. Hijacking incoming calls in networks with encryption disabled

There are numerous types of impersonation attacks that can take place in a communication network. These include: Username/password, Access point, Mobile node, and IP address. IP address impersonation is particularly relevant, as this not only applies to mobile phone networks but to Internet use as well. The fourth generation of mobile telecommunication technology 4G will feature full IP packet data transmissions. To comprehend the security problems with IP, first the types of IP and the architecture it is used in must be understood.

2.2 Mobile IP Protocol

IP mobility allows an mobile node to migrate from one location to another, which primarily occurs when a node physically changes its point of attachment to the network, and yet remains in continuous communication with the network [2]. This paradigm was previously unavailable as the TCP/IP protocol only operates with a static IP address in the sense that it does not change during the time of the connection. Portability was achieved by disconnecting a mobile device from its current point of attachment and reconnecting at another network connection. Mobile IP has addressed this issue by providing two IP addresses:

- The home agent, which is a static address and resides on the home network.
- Care of address, which relates to the mobile node's current location.

When corresponding nodes attempt to communicate with a mobile device at its current physical location, it transmits a packet to the home agent. The home agent then encapsulates the packets with IP in IP encapsulation, which adds an outer header to the data packet with and new destination address, and tunnels them to the care of address where the mobile node decapsulates the packets (Figure 1).

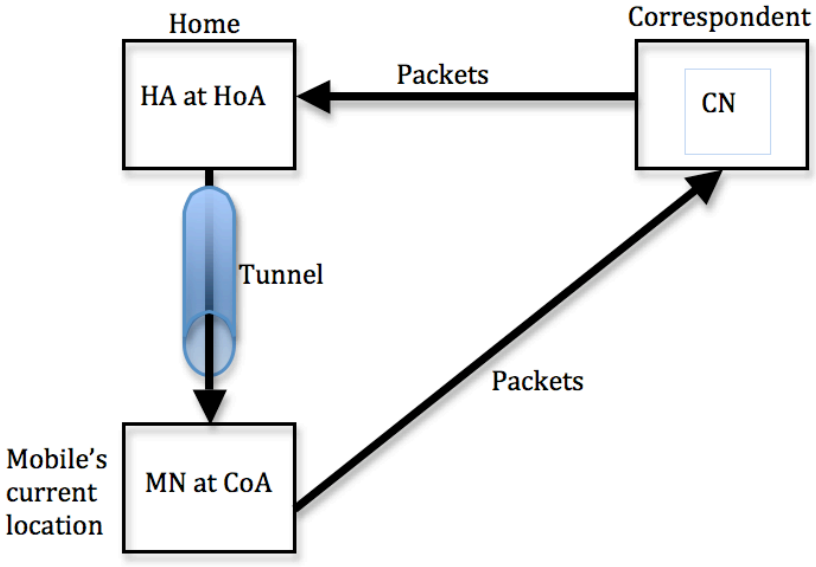

Figure 1. Packet routing in Mobile IP

The home agent is the only node to record the current location of the mobile node allowing network transparency when the mobile device moves from one point of attachment to another. Thus any communication with the mobile node must be via the home agent, however the mobile device can communicate directly with the corresponding nodes but the return address references the home agent.

Route optimisation, introduced in IPv4 as an add on, allows the corresponding node to also keep track of the mobile device's current care of address allowing it to cut out the middle man and send packets directly to it. Route optimisation is an integrated feature of IPv6 [1]. IPv4 addresses are 32 bits in length, however the number of devices requiring an IP address has grown at an exponential rate leading to a shortage of available addresses and to the development of IPv6. IPv6 addresses are 128 bit and consist of a 64 bit routing prefix and 64 bit interface identifier. IPv4 uses an extra node called a foreign agent, which the mobile attaches itself to. The foreign agent acts like a home agent within the foreign network forwarding data packets to the mobile node and dealing with matters of address auto-configuration and neighbour discovery. There is no demand of a foreign agent in IPv6, as the mobile node is able to operate most or all of the jobs done by an IPv4 foreign agent. In IPv6, each node is required to be able to operate the functions of address auto-configuration and Neighbour Discovery [2].

2.3 Protocols within Mobile IP

2.3.1 Neighbour discovery (ND) and Router discovery (RD)

IPv6 neighbour discovery (ND) and router discovery (RD) functions are focused on in [18]. Every node can potentially be untrustworthy and launch a range of attacks such as denial of service, man in the middle and masquerade. Cryptographically generated addresses (CGA)[11] and address-based keys (ABK)[10] are useful in bringing security to the IPv6 local link.

2.3.1.1 Neighbour Discovery

Neighbour discovery [18] in IPv6 provides nodes with the ability to discover the presence of other local link nodes and their link layer addresses. They are also used for detecting when a local node becomes unreachable, fixing duplicate addresses and for nodes to be redirected to more appropriate routers.

There are three main security properties for neighbour discovery:

- **Address Resolution –** The node sends a neighbour solicitation message to a multicast address to learn the link layer address of another node. If the node is present it will reply with a neighbour advertisement message. Messages may be protected with IPSec AH [12]. However key distribution problems mean the system fails to operate. Authentication keys should be strongly bound to IP addresses to allow nodes to verify that the user of an authenticated packet belongs to the appropriate address.
- **Neighbour Unreachability Detection (NUD) -** This is a procedure, which is invoked when there is a long delay or the node stops receiving replies from a peer node. A neighbour solicitation is sent to the peer node and waits for a neighbour solicitation. If after a few attempts this has not happened then the neighbour cache entry is deleted and the address resolution protocol is triggered if necessary.
- **Duplicate Address Detection (DAD)** - When a node wishes to use a new address it must first check to see if anyone else is using it. To do this it sends a neighbour solicitation to the local link with a message asking if this address is already in use? If the address is in use then a neighbour advertisement is sent in reply, and the process repeats its self until no reply is sent at which point the address can be used.

2.3.1.2 Router Discovery

Router discovery [18] allows nodes to learn the identities of the routers attached to the link and their capabilities. It also provides nodes with globally routable address prefixes. Router advertisements are also periodically multicast.

2.3.2 Auto-configuration

This combines the processes already mentioned into a method of providing initial boot time to an IP host. Stateless auto configuration [18] begins by performing duplicate address detection. The second phase is router discovery and finally the host uses DNS server discovery to learn the identities of local Domain Name System (DNS) servers.

2.4 Problems with Mobile IP

Problems in the implementation of Mobile IP occur when the Mobile Node is away from home [19]. Routers installed on the Internet are beginning to perform IP filtering in order to eliminate IP spoofing, effectively becoming a firewall. The IP spoofing attack [5] is the forgery of source IP addresses. Some programs base authentication on the source IP address of packets they receive. Therefore, if an external machine fakes its IP address it could gain unauthorised access to machines within an organization. Therefore, those routers filter the incoming packets of the external connection. If the packets, which arrive from the external connection to the organization, have the IP address of any internal machine, they drop them. With this scenario, Mobile Nodes cannot communicate with its home network when away from home, apart from the registration mechanism.

2.5 Route Optimisation Using Binding Updates

When the mobile node receives a tunnelled packet from the home agent, it initiates the route optimisation protocol, which allows the mobile node to send a binding update (BU) message to the corresponding node [3, 2]. The binding update contains the mobile node's home address and current care-of address. The correspondent stores this information in its binding cache, which acts as a routing table. The binding update informs the correspondent that packets destined for the home address should be rerouted to the current care of address. Finally, the correspondent confirms the receipt of the binding update and replies with a binding acknowledgement.

Route optimisation is voluntary and the mobile or the correspondent is able to refuse to participate, continuing to communicate via the home agent. However sending packets via the home agent can be very inefficient and so the optimisation is recommended. Route optimisation typically works as shown in Figure 2.

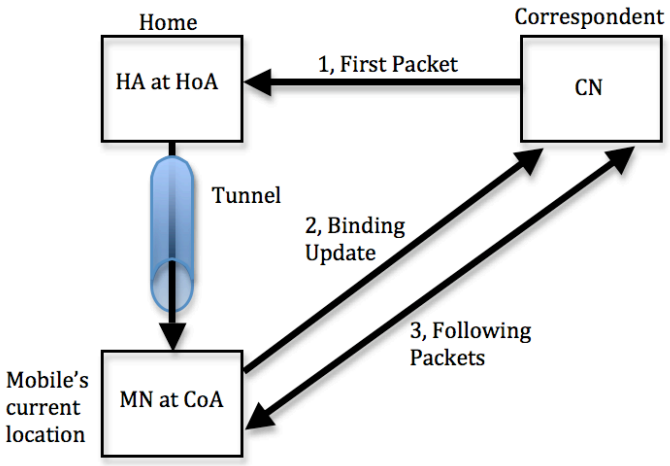

Figure 2. Binding update route optimisation

Once the binding update has taken place, the mobile and corresponding nodes are able to send packets to each other directly. The packets from the mobile to the correspondent contain a header field called the *home address option*, which informs the correspondent that the packet is to be considered as originating from the home address node instead of the actual care of address found in the source field. This is to allow for better compatibility with ingress filtering [3]. The packets from the correspondent to the mobile contain a *routing header*, which informs the mobile that packets destined for the care of address, are really intended for the home address, effectively creating a tunnel between the nodes. Every few minutes, the mobile needs to send another binding update to refresh the binding cache entry even if the care-of address has not changed.

2.6 Summary

Wireless communication technologies have come along way with improvements in every generational leap. As communications evolve so do the system architectures, models and paradigms. Improvements have been seen with jump from 2G to 3G in terms of security. Yet security issues persist and will continue to plague mobile communications in to the leap to 4G if not addressed. 4G will be based on the transmission of Internet packets only using architecture known as mobile IP. This will feature many advantages however security is still a fundamental issue to be resolved. One particular security issue involves the route optimisation technique, which deals with binding updates. This allows the corresponding node to by-pass the home agent router and communicate directly with the mobile node. The home agent has a static address while the mobile node's address changes every time it moves to a new location with a new point of attachment. The home agent keeps track of the mobile node's current address, so that if a correspondent does not know it, it may send the packets to the home address, which will forward the packets to the mobile node. By bypassing the home address with the binding update route optimisation, the speed of the delivery of packets will increase.

The next chapter will look at how binding updates are vulnerable to attacks, what types of attacks may occur and the results of these attacks.

Chapter 3 Analysis of Security Vulnerabilities of Binding Updates

If the binding update protocol is implemented it would create serious new security vulnerabilities as they are not encrypted, authenticated or protected in anyway. An attacker C can send a false binding update to a correspondent, claiming to be a mobile with home address A, making the correspondent B, redirect all packets intended for the legitimate home address A to the attacker allowing him to intercept them. This attack is shown in Figure 3.

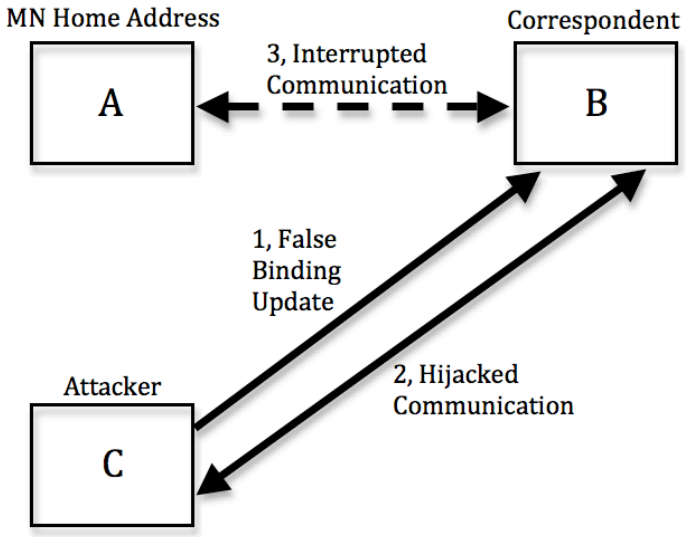

Figure 3. False binding update

The attacker can also spoof legitimate data packets by inserting a false home-address option in them [3]. This enables the attacker to hijack existing connections between the mobile's home address and the correspondent, and to open new ones pretending to be the mobile. The attacker can also redirect the packets instead of intercepting them, preventing the appropriate nodes from communicating with each other. The impact of this attack has been reduced with the introduction of return routability and Cryptographically Generated Addresses as discussed in chapter 4.

End-to-end data protection, such as IPSec [12] or SSL, prevents most of the attacks but not denial of service (DoS). These attacks are serious as IPv6 addresses can be any node anywhere on the Internet. All the attacker needs to know is the IPv6 addresses of the mobiles home agent and the corresponding node. Since there is no visible difference between a mobile home address and a stationary IPv6 address, the attacks work as well against stationary Internet nodes as against mobile ones. The possibility of these attacks caused the Internet Engineering Task Force (IETF) to halt the Mobile IPv6 standardization process until a solution for authenticating binding updates was found [3]. It is believed that deployment of the protocol without security could result in a breakdown of the entire Internet. Obviously, the solution is to authenticate the binding updates. Typical authentication mechanisms need to work between any mobile Internet node and any correspondent involving a trusted online server or a

public-key infrastructure (PKI). The problem is that the there is no authentication infrastructure that currently exists that could be used for global authentication between any two IPv6 nodes and creating such an infrastructure is unrealistic. Hence unconventional authentication methods must be considered.

3.1 Risks of Unauthenticated Binding Updates

Accepting an unauthenticated binding update leaves the node vulnerable to several attacks. These include denial of service attacks where a false binding update is accepted that sets the address of a service to a non-existent address, effectively disrupting communication [15]. A variation of this attack is when the attackers own address is set as the service allowing the provision for a maliciously modified service. For this reason nodes on the Internet are recommended to use a form of authentication for binding updates.

3.1.1 Spoofing Redirect Messages

An attacker can send a redirect message to a host and claim that it came from the current first hop router by using the link- local address of the source address of the message. This will redirect packets to the link layer address and will continue as long as the attacker responds to the neighbour unreachability detection probes [18].

All the three types of threats above can result in the following consequences:

1. The interception of packets.
2. The prevention of communication by redirecting the packets to the wrong address.
3. Networks node can be flooded by the redirection of large amounts of data.

3.1.2 Resource Exhaustion and Other Denial of Service Attacks

There are two types of denial of service attacks in mobile IP. The first is resource exhaustion where there is a limited amount of resources such as bandwidth or processing power and this is consumed by the attack leaving the victim unable to use them [15]. The second type of denial of service attack is that of forged binding updates, giving the correspondent the wrong home and care of addresses resulting in packets being unable to be routed to the correct destination, effectively cutting the mobile from the network.

Binding updates in mobile IPv6 is an optimisation and the mobile node can still communicate with the correspondent if their acceptance is refused. However, performance will suffer, as the packets will have to be routed via the home agent. This suggests that correspondents can defend themselves against resource exhaustion attacks by stopping the processing of binding updates when it is flooded with an overwhelming amount of binding updates that fail the cryptographic integrity check. It may be possible

for the correspondent to come to the conclusion that not accepting any binding updates at all is more efficient than spending resources on checking false binding updates. Nodes that are willing to respond to anyone, expending valuable resources without assessing the senders intentions, risk being vulnerable to a resource exhaustion attack.

A possible solution is to split the authentication protocol into two phases [15]. The first phase uses a low level of security and consumes very little resources. The second phase uses a high level of security and requires more resources to provide it. The second phase can only be entered if the first phase completes successfully. This process is used in the CAM protocol.

3.2 Risks of Unauthenticated Binding Acknowledgements

Accepting a forged binding acknowledgment is considered to be less serious than that of binding updates. The consequences of this is that the correspondent will fail to create an up to date binding for the victim, the previous binding will expire and the correspondent will revert back to using the home agent to communicate with the mobile. This will allow a degraded form of communication.

However, a more serious problem is forged binding acknowledgements that appear to come from the home agent of the victim [15]. If the mobile node attempts to send a binding update to the home agent but the message is blocked by an attacker and then a false binding acknowledgment is sent, the home agent will lose contact with the mobile node. Even if the attacker cannot block the update, repeated sending of acknowledgements may disrupt communication if an update is lost, preventing its retransmission.

3.2.1 Inducing Unnecessary Authentication

An attacker can exploit the binding update protocol by sending spoofed IP packets to the mobile that appear to come from different correspondents [1]. Entries in the binding cache will be useless and the attacker can execute the binding update protocol unnecessarily draining the mobiles resources. In this case the stronger the cryptographic authentication protocol the more vulnerable it is.

3.2.2 Consuming Authentication Resources

PKI are resource intensive and so this can be exploited by an attacker to drain the mobile of them by launching an attack that floods it with packets that need to be authenticated [1].

3.3 Risks of Not Authenticating Home Agents

The consequences of an attacker posing as a home agent are serious in terms of confidentiality. The attacker has three options:

1. Prevent transmission of packets to the care of address resulting in a DOS attack.
2. Eavesdrop on data packets breaching the users confidentiality.
3. Modify transmitted packets for the attackers own malicious purposes.

2 and 3 are considered man in the middle attacks. To avoid these attacks the home agent should go through an authentication procedure [15]. It is not sufficient that the home agent's address be statistically unique, as this does not bear on how it will behave. This suggests that techniques which derive addresses from public keys are not sufficient in this case, it only shows that the address is unused by another node. Even IPSec may not be sufficient as the certificate-based key management associating keys with IP addresses does not provide assurance that the home agent is trust worthy.

3.3.1 Malicious Last Hop Router

The attacker can masquerade as a last hop router by replying to a router solicitation or multicasting router advertisements. If it is selected then it will be able to redirect all traffic passing through it [18].

3.3.2 Bogus on Link Prefix

An attacker can send a router advertisement message specifying that some prefix is on-link. The result is that it will never send a packet for that prefix to the router. The host will try to perform address resolution by sending neighbour solicitations but will not get a response, denying service to the attacked host [18].

3.4 Denial of Service Attacks against Mobile Nodes

3.4.1 Reflection and Amplification

It is possible for an attacker to trick nodes in to sending the same number of packets, or more to a target. It is possible that packets can be sent on a looping path allowing them to be reflected multiple times, amplifying the amount of attack data [1].

3.4.2 Piggybacking and Jitter

Control messages such as binding updates can be combined with other message and is called piggybacking. This can cause delays however as the files are larger, taking longer to transmit and also require more processing power. If this is combined with the authentication of asymmetric cryptography or digital signatures then the processing will incur noticeable delays called jitter. This can reduce the quality of service of the network and also some applications are sensitive to transmission delay such real time voice communication [15].

3.5 Denial of Service Attacks against Home Agents

An attacker may have two different home addresses and then send binding updates to bind them to each other as care of addresses, creating a routing loop [6]. Packets caught in the loop will eventually time out but the loop will create traffic amplification, which may affect the quality of service of the network, even causing denial of service. For each packet the attacker sends in to the loop the home agents send many. To protect against this attack home agents should only act on behalf of trustworthy mobile nodes, which it knows.

3.5.1 Neighbour Solicitation/Advertisement Spoofing

An attacker can spoof neighbour solicitations and advertisements by giving the address of a potential victim in the source or target link layer address respectively. This will cause packets for legitimate nodes to be sent to another link layer address [18].

3.6 Risks of Not Verifying the Care-Of Address

Security protocols can prevent dos attacks by verifying that packets sent to a mobile's claimed care of address reach a willing participant of the protocol preventing redirection attacks [5]. If security protocols do not authenticate the care of address an attacker may be able to intercept packets sent to it, complete the protocol and then flood the unwilling care of address with data.

3.6.1 Bogus Address Configuration Prefix

An attacker can advertise a false subnet prefix. The host executing auto-configuration will use the prefix to construct an address resulting in return packets never reaching the host [18].

3.7 Stateless Protocols

Stateful protocols can expose participants to denial of service attacks because the correspondent must store a separate state for each mobile it is in communication with and keep track of all the keys in use. The attacker can initiate the protocol many times causing the host to store large amounts of unnecessary protocol states. This can be prevented with additional memory and optimised state storage management or using a stateless protocol [1]. The correspondent would not have to store a separate state for each mobile it is in communication with as it will only have to store a single periodically changing randomly generated master secret. This allows the correspondent to remain stateless until the mobile has been authenticated.

3.7.1 Duplicate Address Detection DOS

When a host enters a network and uses stateless address auto-configuration to obtain an address, it is possible for an attacker to respond to every duplicate address detection attempt claiming that it owns that address. This prevents the host from getting an address [18].

3.7.2 Neighbour Discovery DOS Attack

An attacker can continuously send bogus addresses with a valid subnet prefix to the target network but with an invalid suffix [18]. A host entering the network will be unable to obtain a neighbour discovery service from the last hop router, as it will be busy resolving the bogus addresses.

3.7.3 Parameter Spoofing

An attacker can duplicate a valid router advertisement but change the parameter values to disrupt traffic. It could claim that the network uses DHCP for address configuration but as the host attempts to contact the non-existent DHCP server it will be unable to get any usable IP addresses [18].

Many of these attacks can be avoided with the use of authentication mechanisms, however even authenticated binding updates can be used to amplify a packet flooding attack.

3.8 Threats from a Lack of Authentic Location Information

An attacker may be able to misinform correspondents about the mobile nodes location preventing packets from reaching the correct destination. This is a compromise in confidentiality if the packets are redirected to the attacker, otherwise is classified as a denial of service attack. To maintain the attack, a new binding update has to be sent every few minutes to refresh the binding cache entry of the correspondent. Attackers can be anywhere on the network and all nodes are potential targets.

In an effort to make mobility transparent in MIPv6, all node support correspondent functionality and mobile node addresses are indistinguishable from stationary addresses.

Public servers and those using stateless auto-configuration [1] are most vulnerable because an attacker needs to know the IP addresses of both communicating nodes to send a false binding update. There are two types of attack:

- **Active attack** - easier for the average attacker. It can initiate binding update protocol at any time.
- **Passive Attack** - requires the attacker to wait for messages to be sent by the target node.

Security threats of lack of authentic location information are divided in to two types:

1. Redirection
2. Denial of service

3.8.1 Redirection (Bombing)

Heavy data streams between nodes can be redirected to a target by sending a false binding update to amplify a packet flooding denial of service attack [1]. Binding update authentication prevents this attack, however the attacker may provide false location information about its own location. This would allow it to direct the data stream towards a node and it may be possible to spoof the acknowledgments. This is a serious threat because the victim cannot prevent the attack by not sending or accepting binding updates. A technique of this nature is particularly effective with a distributed denial of service attack as the effect is amplified.

3.9 Summary

There are a variety of security vulnerabilities with binding updates, which include the interception of data packets, which would allow an attacker to eavesdrop on its contents breaching the users confidentiality or to modify transmitted packets for the attackers own malicious purposes. Other possible vulnerabilities with mobile IP include address spoofing, redirection and denial of service attacks. For many of these attacks all the attacker needs to know is the IPv6 addresses of the mobile's home agent and the corresponding node.

The next chapter will look at the possible security solutions, which can be used to address the vulnerabilities mentioned in this chapter. It will look at cryptography, different types of authentication architectures and also security protocols specifically designed for the protection and authentication of binding updates.

Chapter 4 Analysis of Security Protocols and Architectures for Binding Updates

The previous chapter examined the different types of security vulnerabilities concerning binding updates. This chapter will look at the various types of security that can be utilized in an effort to make optimised communication less vulnerable to attack. The chapter begins by looking at cryptography, especially elliptic curve, which has many advantages over traditional public key systems. Cryptography allows communication between participants to remain secret even if it is intercepted by scrambling the data with a mathematical algorithm. Keys are strings of data that contain the necessary data for the algorithm to reconstitute the data. Only the use of the correct key will allow the decryption to occur. Authentication techniques are then examined which are designed to identify that the user attempting to use the system is who they claim to be. Authentication protocols, which have been especially designed for mobile IP binding updates, are looked at and finally location privacy is examined as a potential solution to one of the problems of existing security protocols.

4.1 Encryption

4.1.1 Symmetric Vs Asymmetric

There are two types of encryption, symmetric and asymmetric [4]. Symmetric encryption uses a single key to encrypt and decrypt data. In contrast asymmetric encryption uses two keys, one to encrypt the other to decrypt called the public and private keys. The advantages of using a symmetric key system, such as Data Encryption Standard (DES), are that it is fast and not very resource intensive. However the problem lays in how to securely distribute the keys. This is one of the strengths of an asymmetric key system, such as Rivest, Shamir and Adleman (RSA), where the public key is sent to the user wishing to communicate and uses it to encrypt the data. If the key is intercepted, the attacker will only be able to encrypt data not decrypt it. This is a different problem and is solved with authentication. Asymmetric key systems are resource intensive and can consume 100 – 1000 times more processing power than symmetric keys. The ideal would be to combine the two to securely transmit a symmetric key with an asymmetric one.

4.1.2 Elliptic Curve Cryptography

ECC stands for Elliptic Curve Cryptography [20]. It provides an alternative to older public-key encryption such as the RSA system and has many advantages. ECC devices require less storage, less power, less memory, and less bandwidth than other systems. This allows the implementation of encyption in platforms that are constrained, such as wireless devices, handheld computers, smart cards, and thin-clients.

Current key-size recommendation for older public key schemes is 2048 bits, ECC key offers the same level of security with 224-bits. As older keys sizes increase ECC still remains small, which is a long-term benefit. The comparison of different key sizes in different encryption algorithms is shown in Figure 4.

Elliptic curves can provide versions of public-key methods that can be faster and use smaller keys, while providing an equivalent level of security. Their advantage comes from using a different kind of mathematical *group* called discrete logarithms as opposed to exponentiation.

NIST guidelines for public key sizes for AES

ECC KEY SIZE (Bits)	RSA KEY SIZE (Bits)	KEY SIZE RATIO	AES KEY SIZE (Bits)
163	1024	1 : 6	
256	3072	1 : 12	128
384	7680	1 : 20	192
512	15 360	1 : 30	256

Figure 4. Comparison of different public key sizes [20]

Discrete log is the inverse operation of exponentiation and is far more complex but only when the modulus is very large. Diffie-Hellman and other public key methods can work with elliptic curves, which can allow for tailor made solutions.

The basic principle of ECC is to add any two intersecting points on the curve, which equates to a third point. The rule for adding **a** and **b** is to draw a strait line through them to find the third intersecting point **b**, then draw a vertical line through these points to find another intersecting point **c** on the curve [7]. The sum of **a** + **b** = **c.** Each point along the curve is an integer co-ordinate and only four intersecting points are shown in Figure 5. A functioning curve would consist of many points called P. O is the origin and acts as a third element in a co-ordinate and its addition is analogous to multiplying by 1.

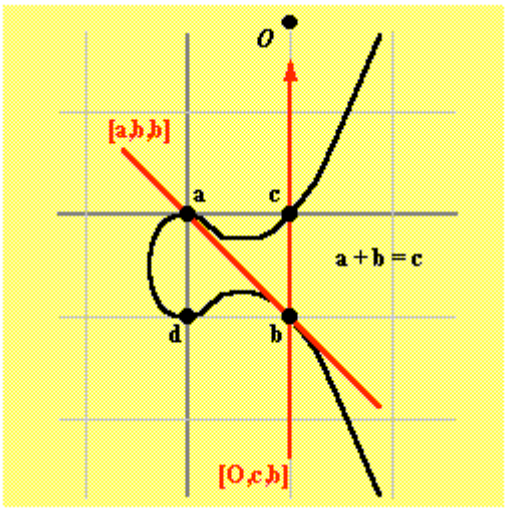

Figure 5. ECC - The third point c is calculated from the addition of a and b [7]

Only the points of integer coordinates (x, y) are considered for the calculation. The curve arithmetic is performed by the modulus of p, where p is a large prime number or a large power of two. The elliptic curve contains N points, where N is almost equal to p, k is a small number and q is a prime number.

N = k * q.
The calculation for each elliptic curve changes because each curve is different. This is part of what make this algorithm difficult to break. It has smaller key sizes than other public key systems with comparable strength however there is a set up cost in the construction of the curve itself.

According to [21], Elliptic curves have solved the problem of identity-based encryption and are used to construct identity based encryption and key agreement protocols.

4.2 Authentication Protocols in Mobile IP

Encryption allows data to be kept private even if it is intercepted. But how do the participants know that the data they receive is in fact authentic and has kept its integrity? Authentication techniques are designed to verify the identity of the users attempting to use the system and to make sure that the integrity of the data is in tact.

4.2.1 Key Exchange Protocols

4.2.1.1 Needham-Schroeder

Needham-Schroeder Protocol [22, 23] is based on symmetric and public key based protocols for key establishment and transport. Their protocols satisfy a number of properties, including mutual

identification of the participants, key authentication, and the establishment of a shared key. This symmetric key protocol forms the basis for the Kerberos authentication protocol.

A trusted third party acts as the KDC (Key distributed centre), which authenticates the users and provides secure exchange of the session key.

4.2.1.2 Kerberos

Kerberos [8, 24] is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography.

Kerberos is based on the key distribution model developed by Needham and Schroeder [22]. Kerberos was designed to eliminate the need to demonstrate possession of private or secret information (the password) by divulging the information itself. Kerberos is a network authentication protocol, which utilizes symmetric cryptography to provide authentication for client-server applications. The core of the Kerberos architecture is the KDC (Key Distribution centre). The KDC stores authentication information and uses it to securely authenticate users and services.

This authentication is called secure because it:

- Does not occur in plaintext
- Does not rely on authentication by the host operating system
- Does not base trust on IP addresses
- Does not require physical security of the network hosts

The KDC acts as a *trusted third party* in performing these authentication services. Due to the critical function of the KDC, multiple KDC's are normally utilized. Each KDC stores a database of users, servers, and secret keys. Kerberos clients are normal network applications, which have been modified to use Kerberos for authentication.

Because the KDC's store secret keys for every user and server on the network, they must be kept completely secure. If an attacker were to obtain administrative access to the KDC, he would have access to the complete resources of the Kerberos realm. Kerberos tickets are cached on the client systems. If an attacker gains administrative access to a Kerberos client system, he can impersonate the authenticated users of that system. Kerberos uses the DES algorithm for encryption. Kerberos also supports the cyclic redundancy check (CRC-32), Message-Digest Algorithm (MD4, MD5), and the Data Encryption Standard (DES) algorithms for checksums. Kerberos implementations are free to add additional algorithms for encryption and check summing.

4.2.1.3 Diffie-Hellman Key Exchange

This is a method of establishing a shared key across an insecure channel. Two large numbers n and g are selected and made public. Next user a and b each select their own secret number x and y respectively. User a sends g^x mod n to user b, who will then have enough information to calculate the shared key g^{xy} mod n. It then sends user a g^y mod n allowing for it too to construct the shared key.

Security protocols typically employ an authentication phase followed by a protected data exchange. In some cases, such as TLS (Transport Layer Security), these two phases are tightly integrated, while in other cases, such as EAP (Extensible Authentication Protocol) and Kerberos, they are separate and often implemented in different endpoints [25].

Public Key Infrastructure (PKI) could be used for decentralized authentication, without requiring an on-line authentication server. However, current PKI's are not that well suited to authorization and a centralized server is needed. Often authentication and authorization information are combined in a single authentication, authorization and accounting (AAA) server [14].

4.2.2 Authentication Mechanisms

4.2.2.1 Challenge Response Protocol

The Challenge response protocol [4] provides a basic form of authentication. It works by one user sending a challenge to another user and if they are able to encrypt it with the shared key and send it back in reply, proving they have the key. Then the challenge occurs again in the opposite direction. This takes a number of steps and optimisations have been tested but are subject to security flaws such as reflection attacks. The optimisation involves combining steps of sending a challenge and reply to the previous one simultaneously. The reflection attack, for this, works by sending a challenge and receiving the encrypted challenge in reply along with the sender's new challenge. The attacker can open a new dialogue and transmit the challenge it had previously received. The idea is to get the victim to do the encryption on behalf of the attacker, which occurs. The attacker receives the encrypted challenge and then goes back to the original communication and sends the encrypted challenge to the victim "proving" that the shared key is known.

4.2.2.2 Mutual Authentication

This is similar to the challenge response protocol where both parties authenticate each other with encrypted challenges however in the second message the shared key is transmitted and the challenge is replied to by encrypting it with the shared key [4].

4.2.2.3 Digital Signatures

In addition to authentication, digital signatures can be used for non-repudiation. Messages are signed with the users private key, uniquely tying it to the data. Message m is encrypted with user A's private key Ka^-. This can then be encrypted again with user b's public key for confidentiality. Kb+(m,Ka-(m)). M is sent twice for the purposes of integrity checking, however this is costly and alternative is to use a message digest, i.e a hash. The recipient can decipher the hash and create a hash of the message and compare them to see if modification has taken place [4].

As long as the private key remains secret the protocol is secure. However if the key is lost or changed then repudiation may occur. This can be strengthened with the use of time stamps and even a tracking central authority.

4.2.2.4 Hash Values

Hash values are primarily used for accessing data or for security. A hash value (or simply hash), also called a message digest, is a number generated from a string of text. The hash is substantially smaller than the text itself, and is generated by a formula in such a way that it is extremely unlikely that some other text will produce the same hash value [4]. Hashes can be used for integrity checks by sending a message with its corresponding hash. The recipient creates a hash of the message and compares it to the received hash to see if any modifications have taken place. If the values are equal the data is unmodified.

4.2.3 Authentication Architectures

4.2.3.1 AAA - Authentication, Authorization and Accounting

Within the Internet, a client belonging to one administrative domain (called the home domain) often needs to use resources provided by another administrative domain (called the foreign domain). An agent in the foreign domain that attends to the client's request (called the "attendant") is likely to require that the client provide some credentials that can be authenticated before access to the resources is permitted [14]. These credentials may be something the foreign domain understands, but in most cases they are assigned by, and understood only by the home domain, and may be used for setting up secure channels with the mobile node.

Originally AAA protocols [26] were designed for dial up users accessing an ISP securely. AAA defines a framework for coordinating individual security and network components such as authentication of users, authorization to access services and accounting of services used for billing, across multiple network technologies and platforms. An AAA server communicates with an AAA client to provide

distributed AAA services. The AAA server holds a database of user profiles to aid in authenticating and validating the end users prior to granting network access. The end user must possess unique identification such as a password, a cryptographic key or biometric data this is then compared to the user associated data stored in the database. If the data is a match the user gains access to the network, other wise access is denied.

Once access is granted authorization defines which services the user is allowed to access. This may include the provision of an IP address, which applications may be accessed and which protocols are supported. Accounting monitors the users use of the network and collects information about resource consumption to aid in billing and auditing.

4.2.3.2 Radius Authentication Architecture

RADIUS [13] servers are responsible for receiving user connection requests, authenticating the user, and then returning all configuration information necessary for the client to deliver service to the user. A RADIUS server can act as a proxy client to other RADIUS servers or other kinds of authentication servers. Transactions between the client and RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, any user passwords are sent encrypted between the client and RADIUS server, to eliminate the possibility that someone snooping on an unsecured network could determine a user's password. The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the user name and original password given by the user, it can support Point-to-Point Protocol (PPP), password authentication protocol (PAP) or Challenge-Handshake Authentication Protocol (CHAP), UNIX login, and other authentication mechanisms.

The AAA protocol with the widest deployment is Radius (Remote access dial-in service) [13]. The principles are the same as AAA with a client server based operation. Figure 6, shows a typical RADIUS configuration and the relationship between components. The user connects to the RADIUS client supported network access server (NAS). At the time of design the user would use a dial in service. The client collects the user name and password from the user and uses UDP/IP to transmit an encrypted access-request message to the RADIUS server. The RADIUS server compares the user name attribute to the entry stored in the database. If there is a no match then the server returns an access-reject message to the client with a message explaining the reason for rejection. The client then informs the user. However if a match is found then the server returns an access accept message to the client with configuration information such as an IP address to complete the connection.

Radius provides network tunnelling, which is compulsory in the dial-up virtual private network (VPN), using a protocol such as the layer 2 tunnelling protocol (L2TP). Tunnelling is the forwarding of packets from the home agent to the mobile node using IP within IP encapsulation.

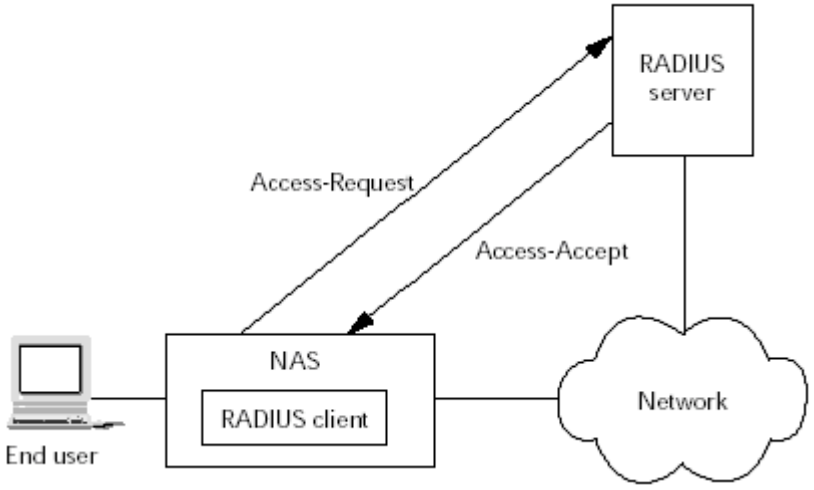

Figure 6. RADIUS authentication architecture [26]

4.2.3.3 Diameter Protocol

Radius was originally designed for small network devices requiring simple server based authentication with only a few end users. AAA services are now used by thousands of concurrent end users and over a variety of technologies. This has become a burden for the protocols capabilities and so a new AAA protocol was developed.

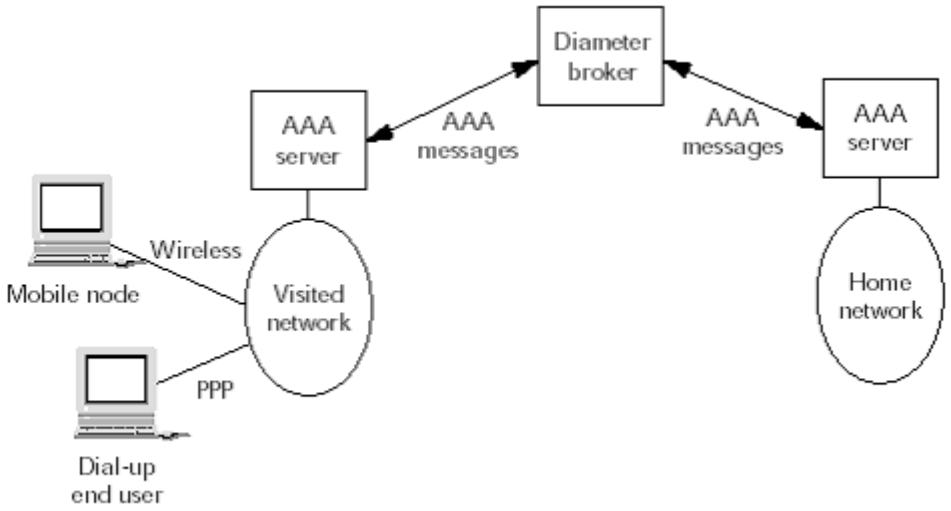

Figure 7. Mobile IP AAA with Diameter [26]

Diameter [26] is peer based and light weight, offering scalability for the introduction of new policies and services for emerging network technologies such as roaming and Mobile IP. It employs many of the same mechanisms as RADIUS and tries to correct its limitations. Figure 7 shows how a Diameter broker allows AAA service delivery from Mobile IP users, in a foreign network, to access resources in the home network. The Diameter server in the visited network communicates securely with the broker

as a peer to execute AAA functions. The broker has the ability to act as a certificate authority allowing for scalability in contrast to the use of shared secret keys.

4.2.3.4 IPSEC

IPSec is short for **IP Sec**urity. The basic definition of IPSec [27] is a set of protocols developed by the IETF to support secure exchange of packets at the IP layer. IPSec has been deployed widely to implement Virtual Private Networks (VPNs).

The IPSec [12] security protocol operates within the network layer and provides authentication, integrity, access control and confidentiality within its cryptographic security service. Two protocols are provided for traffic security, IP Encapsulating Security Payload (ESP) and IP Authentication Header (AH) [28]. Both ESP and AH focus on protecting the IP datagram from eavesdropping and modification of the message. The Authentication Header provides authentication by adding authentication information to each IP packet to provide security. The ESP protocol provides confidentiality to packets transmitted if it is required. However AH or ESP hide the source and destination IP addresses of the communicating parties exposing their location. Because IPSec uses shared secret keys, it relies on a separate mechanism for their distribution. Internet Key Exchange (IKE) [29] is specified as the public key based approach to be used for automatic key management. Other distribution techniques may be used such as the key distribution centre based system Kerberos [8]. Security associations are a fundamental concept in IPSec. There purpose is to protect the traffic stream and can be applied to the AH or ESP security protocols.

There are two types of security association (encryption modes), Transport mode and tunnel mode. Transport mode is a security association between two hosts. If transport mode is used with ESP then security is only provided for the higher layer protocols, not the IP header. If it is used with AH then portions of the IP header are protected also. Tunnel mode is a security association used with an IP tunnel (packet encapsulation). There is an outer IP header with the IPSEC processing destination, and an inner IP header with the final destination of the packet. If AH is used with tunnel mode then parts of the outer IP header are protected as well as the entire inner header (and the higher layer protocols). If tunnel mode is employed with ESP then only the tunnelled inner packet is protected not the outer header.

Basically Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. On the receiving side, an IPSec-compliant device decrypts each packet. For IPSec to work, the sending and receiving devices must share a public key. This is accomplished through a protocol known as Internet Security Association and Key Management Protocol/Oakley (ISAKMP/Oakley), which allow the receiver to obtain a public key and authenticate the sender using digital certificates.

However IPSec is not without its limitations [30]. IPSec is used to protect a number of functions including mobility management and source routing. The combination of IPSec and IKE, which are both considered to be relative heavy weight, creates a vicious cycle that is a potential source of various denial-of-service attacks [31].

To avoid or reduce the effect of DOS and redirection attacks; IPSec AH [12] can be used to protect neighbour discovery messages. The problem with IPSec [30] is that the security associations need to be manually configured and with a large number of them this becomes a burden. IKE [29] is an automatic method of dealing with this, however it comes with its own set of problems. Even if an automatic solution existed there would still be problem verifying the ownership of IP address [31].

4.2.4 Miscellaneous Authentication Protocols

4.2.4.1 Ingress Filtering

To prevent nodes sending spoofed packets, a gateway router or firewall can be configured to screen out outgoing packets that are spoofed to appear as if they originated on another network. Ingress filtering limits the choice of false addresses since the source address field of the IP packet header contains the mobiles new address in MIPv6 binding updates.

However ingress filtering [1] must be applied to the attacker's local network, which is out of the control of any potential victim. Another problem is that MIPv6 allows for an alternative care of address sub option, which permits a potential attacker to send a false care of address without spoofing.

4.2.4.2 Return Routability

The Return Routability Procedure gives the correspondent node some reasonable assurance that the mobile node is addressable at its claimed care-of address and its home address. Only with this assurance is the correspondent node able to accept Binding Updates from the mobile node [32].
The return routability test is the most effective way to limit bombing attacks of the mobile's new address. The correspondent only accepts the binding update if the mobile is able to return the hash of a secret value sent in a packet to the new location. The Correspondent sends two tokens to the Mobile Node, one via the Home Agent and one directly. The mobile node hashes these together creating the key and sends it back to the Correspondent. This proves that the mobile can receive packets at the address where it claims to be [1].

Return Routability tests whether packets addressed to the two claimed addresses are routed to the mobile node. The mobile node can pass the test only if it is able to supply proof that it received secret values called "keygen tokens". These are combined by the mobile node into a binding management key, denoted Kbm [32].

Some malicious entities on the correspondent's local network may be able to capture a test packet but the number of potential attackers is dramatically reduced. The return routability test is complementary to CGA-based BU authentication, which does not prevent bombing of the home network [1].

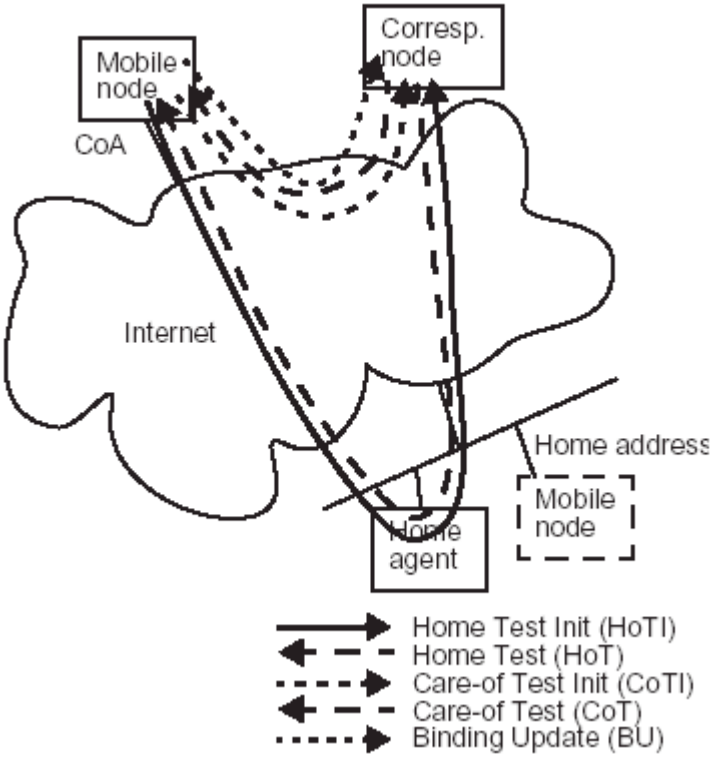

Figure 8. Return Routability [33]

The basic return routability mechanism consists of two checks, a Home Address check and a care-of-address check. The packet flow is depicted in Figure 8.

1) Home Address check:

The Home Address check consists of a Home Test (HoT) packet and a subsequent
Binding Update (BU). The home keygen token is generated by the Correspondent using the following formula: home keygen token = hash(Kcn | home address | nonce | 0) and then sent to the Home Agent. The HoT is tunneled by the Home Agent to the mobile node. The HoT contains a cryptographically generated token, *home keygen token,* which is formed by calculating a hash function over the concatenation of a secret key Kcn known only by the correspondent node, the source address of the HoTI packet, and a nonce [33].

In most cases the HoT packet is forwarded over two different segments of the Internet. It is sent from the correspondent node to the Home Agent, however it is not protected
and any eavesdropper can learn its contents. The Home Agent then securely tunnels the packet to the mobile node.

2) Care-of-Address check:

The care-of check is very similar to the Home check. The Correspondent generates the Care of token using the following formula: care-of keygen token = hash(Kcn | care-of address | nonce | 1).
The only difference is that the packet is sent directly to the care-of-address of the mobile node. It remains unprotected all along the way to the Mobile node, making it vulnerable to eavesdroppers near the correspondent node, on the path from the correspondent node to the mobile node, or near the mobile node. The care of address token is created in a slightly different manner, by using the care of address instead of the home address, and using a different nonce and index number, in order to make it impossible to use home tokens for care-of tokens or vice versa preventing replay attacks [33].

3) Forming the first Binding Update:

When the mobile node has received both the HoT and CoT messages, it creates a binding key Kbm by taking a hash function over the concatenation of the tokens received [33]. This key is used to protect the first and the subsequent binding updates, as long as the key remains valid. A potential security flaw is that the key is available to anyone that is able to receive both the CoT and HoT messages as they are not encrypted as they are sent to the Mobile node. However, they are normally routed through different routes through the network, and the HoT is transmitted over an encrypted tunnel from the home agent to the mobile node. But this may still be susceptible to a coordinated distributed attack.

Return routability only allows a node to verify that there is a node that is able to respond to packets sent to a given address. The check yields false positives if the routing infrastructure is compromised or if there is an attacker between the verifier and the address to be verified [33].

This can be an effective technique to limit bombing attacks. To verify that the mobile truly is in the new location it claims to be, the correspondent sends a packet with a secret value to the mobile's new address and only accepts the binding update if the mobile is able to return the value or its hash. This technique is complementary to CGA based binding update authentication.

The latency associated with Mobile IPv6's return routability [1] test can have an adverse impact on delay-sensitive applications. Early binding updates reduce the impact of latency by already using a new care of address in parallel with testing it. A credit-based mechanism can be used to prevent DOS attacks of Early Binding Updates [34].

With Credit-Based Authorization, a correspondent node measures the "effort" that a mobile node spends for sending or receiving packets, and it grants "credit" for this effort. This limits the data volume that a correspondent node can send to a mobile node's unconfirmed care-of address, and it limits the data rate at which the correspondent node can send this data.

4.2.4.3 Flow Control

One idea to reduce the effectiveness of bombing attacks is to reduce the communication to a single packet when a mobile moves to a new node and then gradually increase the transmission rate. This is a form of return routability, however most transport layer protocol lack TCP compatible congestion control or spoofing of acknowledgements is allowed therefore it is advisable to implement it in the IP layer [1].

4.2.4.4 Delaying Commitment

To protect against DOS attacks, a standard method is to delay commitment of resources until the other party is proved trust worthy. Attacks that exhaust state storage can be prevented by making the protocol parties stateless until proven honest [1]. This can be done by deriving secret values Ka and Kb with a one-way function from a secret value Ni, known only by the correspondent:

Ka=HASH("Ka"/Ni/mobile's home address)
Kb=HASH("Kb"/Ni/mobile's care of address)

4.2.4.5 Limit Damage

A node can protect itself from resource exhaustion attacks by limiting the amount of resources that are allocated for location management. When the allotted resource has been exhausted then communications need to be prioritised [1].

4.2.4.6 Cryptographically Generated Addresses CGA

Cryptographically generated addresses [11] are IPv6 addresses, which are generated by hashing the owner's public key. The address owner uses the corresponding private key to assert address ownership and to sign messages from that address without PKI or some other security infrastructure. 62 bits of interface identifier can be used to store a cryptographic hash of the public key.

Host ID = $HASH_{62}$(public key)

This binds the address to the public key. However ownership claims to the address without public key cryptography can be achieved by generating a chain of hash values [18].

H_N :=$HASH_{160}$(public key | random)
Hi :=$HASH_{160}$(public key|H_{i+1})
host ID :=$HASH_{62}$(H_0)

Hosts avoid public key cryptography by generating a sequence of H index values to H_N. In the case of a collision H_1 of the sequence is revealed. This can be resolved by revealing H_2. It is highly unlikely that a second collision would occur because the HASH is 160 bits instead of 62. If more than the second value of i is revealed then it is probable that an attack is taking place [11].
Brute force and birthday attacks can be discouraged by binding the address to a specific network or hardware address. This can be done by including the routing prefix or link layer address in to the hash.

The CGA binds a users public key to an IPv6 address. The binding between the public key and the address can be verified by re-computing and comparing the hash value of the public key and other parameters sent in the specific message with the interface identifier in the IPv6 address belonging to the owner [35]. A major problem which should be understood is that an attacker can always create its own CGA address but will not be able to spoof someone else's address since the message needs to be signed with the corresponding private key, which is only known only by the legitimate owner.

CGA assures that the interface identifier part of the address is correct, but does little to ensure that the node is actually reachable at that identifier and prefix [35]. As a result, CGA needs to be used together with a reachability test such as return routability, where redirection denial-of-service attacks are a concern.

The CGA offers three main advantages: it makes the spoofing attack against the IPv6 address much harder and allows to sign messages with the owner's private key. CGA does not require any upgrade or modification in the infrastructure [35].

Aim of CGA is to prevent stealing and spoofing of existing IPv6 addresses. The public key of the address owner is bound cryptographically to the address. The address owner can use the private key to sign messages, asserting ownership of the address. However CGA based authentication does not confirm the existence of a node with an authenticated address nor does it verify that it has authorisation for the utilization of the specific subnet prefix [11].
An attacker can create an address by using a subnet prefix and the attackers owns public key, signing messages will prove that the attacker owns the address. What an attacker cannot do is sign messages from someone else's address. The attacker can use someone else's public key to create an address. However they cannot sign messages from it. It may be possible to replay old signed messages by the key owner, however this can be rectified by associating the CGA owner with the address.
The exact address of the sender must be known when a CGA address is used to authenticate messages from the IPv6 node. CGA prevents spoofing of IPv6 Addresses but it does not tell which address is the correct one. CGA is not a complete authentication solution for all purposes but it prevents main avenues of attack [11].

Mobile IPv6 Route Optimisation needs a security solution that can be used between previously unknown peers, without trusted third parties, and on a global scale. Return routability is used for this purpose. Alternative solutions, such as Cryptographically Generated Addresses (CGA) [11] have also been suggested but are not being standardized as the mandatory solution at the moment. In the Hash Based Addresses (HBA) [9] scheme, all mobile nodes that wish to employ MIPv6 Route Optimisation MUST derive the interface identifier part of their addresses by using the one-way hash algorithm SHA1. HBA is simple variant of CGA.

4.2.4.7 ABK - Address Based Keys

Address based keys [10] are an identity based cryptosystem which takes a publicly know identifier, in this case the IP address, and as the public key part of the public/private key pair. The IP address is submitted to a trusted agent called an Identity based private key Generator (IPKG). This generates the private key and returns it via a secure channel. The key pair can then be used. The IPKG generates the key from parameters known to all participants except for a master only known to IPKG.

4.2.4.8 Certified Addresses

An alternative to using addresses as public keys is to use certificates [36] and PKI [22]. Each node generates its own signature key pair and then co-operates with a trusted agent to generate a host identifier. The trusted agent then signs a certificate, which binds the host identifier to the host's public key. This can be in the form of an X.509 certificate. Hosts can then create an IP address by concatenating the rout prefix with the certified identifier.

4.3 Binding Update Authentication Protocols

There are three protocols, which have been created to authenticate binding updates in mobile IPv6:

- Shared Key Protocol [15]
- BAKE/2 [15]
- CAM-DH [15]

All of these protocols have the following goals:

- To prevent the forging of binding updates from malicious nodes.
- To protect nodes from denial of service attacks.
- Make it difficult to exhaust a node's resources.
- Prevent the replay of binding updates.

Each protocol makes different assumptions about their environment and has different levels of processing requirements. Symmetric key is efficient but requires a previously agreed long-term secret

key between the mobile and its correspondents. Bake/2 is also efficient but relies on some routed messages to be protected from a source other than the protocol. CAM-DH uses asymmetric cryptography and as a result requires more processing power however this protocol can be used in scenarios where the others would be inappropriate.

Threats addressed by these protocols:
1. Redirection - This involves a correspondent accepting a false binding update, which sets the mobile's home address to that of another node. This results in a disruption of communication between the mobile and the correspondent and redirects the data to an alternative care of address [5]. The attacker can redirect the packets to himself giving him access to them a providing a threat to confidentiality.

2. DDOS - This works in the same way as the redirection however this time a malicious node sends a forged binding update where the care of address is set to another node. The malicious node then tricks the correspondents to send vast amounts of data to the victim called a distributed denial of service attack [15]. There are different variations to this attack:

- The malicious node begins by sending the correspondent its true care of address. Once its home address has been authenticated it then sends a binding update, which contains the victims care of address.
- The malicious node begins by sending the correspondent its true care of address but continues to send binding updates for it even after it has been allocated to another node. This is a restricted attack, as the attacker's victims have to be using a care of address previously used by the attacker.

3. Resource exhaustion - It is possible to send many invalid binding updates to the correspondent node that would exhaust resources such as processing power trying to verify them, leaving the node unable to cope with other tasks.

4. Replay Attacks - Communication disruption can occur if an attack replays an old binding update the node had sent earlier. If it is accepted then packets will be redirected to the mobile node's old location.

All of these above threats are classified as denial of service attacks.

Bake/2 is based on Return Rouability. Cam is based on Cryptographically Generated addresses and Cam-DH uses a combination of the two solutions. In RFC3775 in 2004 [32], Return Routability was incorporated into the Mipv6 draft. A year later in 2005, Cryptographically generated addresses was also added to the Mipv6 draft in RFC3972 [37]. Since then work has concentrated on optimising the protocols which can be seen in RFC4866 [38] published in 2007, which specifies an enhanced version of Mobile IPv6 route optimization, providing lower handoff delays, increased security, and reduced signaling overhead. Cryptographically Generated Addresses have also been used to counter similar

security issues in the context of SHIM6, a protocol for providing locator agility below the transport protocols, so that multihoming can be provided for IPv6 with failover and load-sharing properties published in RFC5533 [39] in 2009. Other proposals have been made to enhance Cryptographically Generated Addresses such as [40] in 2010. However these are all modifications of the same protocols and shows that there is little need to replace the established security procedures, which effectively carry out their goals.

4.3.1 Shared Key Protocol

This Shared key protocol [15] is used to authenticate binding updates between a mobile and a corresponding node that share a symmetric key. The protocol has the following properties:

- To prevent a malicious mobile from forging a binding update containing another node's home address, it must know the secret key.
- To create a binding update for a care-of address a mobile node needs to be able to receive messages sent to it.
- A mobile does not need to be able to receive messages sent to a particular address to create a binding update that deletes a binding cache entry, however it does needs to know the secret key.

Each correspondent node has a secret key. There is no key distribution mechanism because the key does not need to be shared as the correspondent sends a challenge to the mobile node, which is made from the secret key. Each correspondent node also generates a randomly generated number called a nonce, at regular intervals. A correspondent node uses the same key and nonce with all the mobiles it is in communication with, avoiding the need to generate and store new ones when a new mobile communicates with it. Each nonce value is identified by a subscript. The subscript identifier is communicated in the protocol, so that if it is replaced by subscript +1, the correspondent can distinguish between them and can be checked against the old messages. The correspondent nodes store both the current and the previous nonce, as older values can be discarded. Any messages using the old nonces will be rejected as replays.

Keys can be either a fixed value or regularly updated. An update of the key can be done at the same time as an update of nonce, so that the subscript identifies both the nonce and the key. A correspondent node can generate a fresh key each time that it boots to avoid the need for secure persistent storage.

The protocol operates in following steps:

Step 1 - The mobile node MN establishes a connection with the corresponding node CN and communicates its home and care of address. Refer to List of Notations.

MN ⟶ CN : HoA, CoA

Step2 - The correspondent node sends a binding request to the mobile node containing the challenge (rc), and the subscript identifier serial number (j), which indicates the nonce version (N_j) used to generate the challenge. The challenge can be recomputed from the nonce at any time, relieving the correspondent from storing state data. The challenge (rc) is created by using the Message Authentication Code, MAC, computed on message ($CoA/N_j/1$) with the Corresponding Nodes key K_{CN}.

$CN \longrightarrow MN(CoA):r_c,j$

$Rc = MAC_{KCN}(CoA/N_j/1)$

Step 3 - The mobile node hashes together the shared secret and the challenge to form a session key (K_{BU}), and then uses this session key to authenticate a binding update. The binding update contains the subscript identifier j, so that the correspondent knows which value of the nonce Nj to use to re-compute the session key.

$MN \longrightarrow CN : T_0, HoA, CoA, I, MAC_{KBU}(T_0/HoA/CoA/I),j$

$K_{BU} = Hash(K_h/R_c)$

Once the MAC has been verified, the correspondent creates a binding cache entry. This message contains a tag (T_0) so that it can be distinguished from another variant version of the protocol. The sequence number (I) is also contained in the binding update so that different binding updates sent within the same lifetime of the nonce Nj , can be established in their relative order.

By running the above protocol again, starting at step 2, the correspondent can refresh the binding cache entry for the mobile node when it expires. If the care-of address changes then the mobile node runs the protocol again using the new one, but only if the mobile node is still able to receive messages sent to the old care-of address.

However in that case that there is a change in care of address but the mobile cannot receive messages from the old care of address then an alternative variant protocol is used, the operation of this protocol is as follows:

Step 1 - The K_{BU} key authenticates the binding update sent by the mobile node. To distinguish it from the binding update message of the previous protocol it contains a tag (T_1).

$MN \longrightarrow CN: T_1,HoA,CoA',I',MAC_{KBU}(T_1/HoA/CoA',I'),j$

If the correspondent has a binding cache entry and is able to verify the Message Authentication Code, MAC, then it should mark the binding cache entry as invalid. If the correspondent does not have an existing binding cache entry, then it does not need to verify the MAC.

Step 2, - A new challenge is sent to the new care-of address by the correspondent. The challenge is sent even if the MAC in message 1 is unverifiable. If messages have been lost or state lost has occurred within a node then by the correspondent sending a new challenge to the new care-of address allows the protocol to resynchronise.

$CN \longrightarrow MN(CoA') : r'_c,j'$
$r'_c=MAC_{KCN}(CoA'/N_j'/1)$

Step 3 - Once the MAC has verified the correspondent, the mobile's binding cache entry can be created or updated. This step is the same as the third step of the previous protocol.

$MN \longrightarrow CN : T_0,HoA,CoA',I'',MAC_{K'BU}(T_0/HoA/CoA',i''),j'$
$K'_{BU} = H(K_h/r'_c)$

4.3.2 BAKE/2 Protocol

The BAKE/2 protocol [15] provides a means to establish the shared secret dynamically by extending the shared key protocol. BAKE/2 is only suitable for use in an environment where communications between nodes are protected from eavesdropping by security not supplied by this protocol, such as IPSEC [12] or a physically protected network. This protocol is appropriate when the mobile node has an encrypted tunnel between itself and the home agent. The protocol has a weakness and can be broken by an attacker on the route between the home agent and the correspondent node [59].

The operation of this protocol is as follows:

Step 1 - The mobile's home address and its care of address are both passed to the correspondent by the mobile node.

$MN \longrightarrow CN : HoA,CoA$

Step 2 - The shared secret value (Kh) is generated by the correspondent and assumes that the route is secure when passing it to the mobile node. To test if the mobile can receive messages sent to its home address Kh acts challenge.

$CN \longrightarrow MN(HoA) : K_h; j$
$K_h = MAC_{KCN}(HoA/Nj/0)$

Step 3 - This is the same as step 2 of the shared key protocol where a challenge is sent to the mobile's care-of address by the correspondent.

$CN \longrightarrow MN(CoA) : r_c, j$
$rc = MAC_{KCN}$ (CoA/Nj /1)

Step 4 - The mobile sends an authenticated binding update.

$MN \longrightarrow CN : T_0$,HoA,CoA, I,$MAC_{KBU}$ (T_0/HoA/CoA/i), j
$K_{BU} = H(K_h/r_c)$

The protocol does not defend against an attacker who can monitor the home agent to correspondent node route. However, it can protect the correspondent node against denial of service attacks, which flood it with bogus messages. This prevents resource exhaustion because large amounts of processing power are not used to handle yet to be authenticated messages.

This protocol is suitable for communication between a mobile node and a non mobile server but may not be suitable for communication with a mobile server. This protocol protects against dos attacks where the attacker uses the victim's care of address to redirect high bandwidth traffic to it. Flow control protocols such as TCP do not defend against this attack because the acknowledgements can be forged. The protocol succeeds because it only completes with participants of it.

4.3.3 CAM – Child-proof Authentication for MIPv6

CAM is a unilateral security system for authenticating binding updates in MIPv6 and is described in [16]. The protocol works by incorporating a one-way hash of its public key into the mobile nodes chosen home address. Demonstrating knowledge of the corresponding private key verifies the ownership of the address. This is difficult to falsify given a key pair has to correspond to the hash, yet simply to verify and enforce. This protocol is designed to be used in the absence of an IPSEC [12] implementation.

This system builds upon IPSEC and is lightweight, requiring no manual configuration and utilising minimal message exchanges in comparison to IPSEC and IKE [29]. There is a large interest in mobile IP, which may outpace deployments of IPSEC. IPSEC requires security policy databases and certificates to be installed which increases the administrative burden on all hosts. This extra burden may impede the deployment of MIPv6. This protocol is designed to fill the gap of MIPv6 implementations without IPSEC, which would be otherwise vulnerable to the simplest of attacks.

MIPv4 describes a hash of message fields and a shared secret to form an authenticator however the approach in [16] is fundamentally different in that it does not require shared secrets. It is designed to provide a minimum level of authentication in the absence of IPSEC however [16] recommends using

IPSEC wherever possible between all traffic of the correspondent and the mobile nodes. Some hosts may just be satisfied with using IPSEC only for the binding update part of its capabilities.

When a CAM node is first initialised it creates a key pair, which is stored locally. Then a home address has to be chosen. The 64-bit routing prefix is obtained by listening for local router advertisements. The interface ID is predominantly, in most cases, derived from the interfaces MAC address, as this is economic and globally unique. However the CAM protocol [16] has opted to use a cryptographic one-way hash of the node's public key. A large enough part of the hash has to be selected to render inversion infeasible.

This protocol cannot protect against an attacker that deliberately generates two keys that hash to the same address, as the protocol was not designed with non-repudiation in mind [53]. If a hash of a public key coincides with that of an existing interface ID than the resolution of this conflict can be done with IPv6 duplicate address detection.

This protocol defines its own method of conflict resolution where the node may keep its public key saving it from storing multiple keys. To do this the node generates a modifier, i, which is appended to the public key before the hash is generated. If a conflict occurs then a different modifier is used and the process is repeated until the interface ID is unique. The modifier need not be larger than one or two bits in length.

The mobile nodes binding updates are then ready to be authenticated by the CAM enabled Correspondent. This protocol is easy to deploy as there is no manual configuration and administrative overheads are low compared to IPSEC's need for configuration of databases and certificate deployments.

The CAM protocol can be described by the following notation:
M and C are principals (mobile and correspondent, respectively), A′m is M's care-of address, Ac is C's address, (PKm, SKm) is M's (public, private) key pair, i is the modifier used to resolve name clashes, H(m) is a hash of m, Tm is M's time-stamp, {m}SKm is a signature of m using key SKm, R is the route prefix of M's home address and Am = R, H(PKm, i) is M's home address [53].

$$M \longrightarrow C : A'_m, A_c, A_m, PK_m, i, T_m, \{H(A'_m, A_c, A_m, T_m)\}SK_m$$

The Mobile's public key PK_m is sent together with the modifier, i, to the correspondent C. At first the correspondent compares the mobile time stamp, Tm against its own clock to protect against replays. It verifies that the public key is associated with the address $A_m = H(PK_m, i)$. The correspondent uses the public key to decrypt the hash $H(A'_m, A_c, A_m, T_m)$, generates its own and compares the two. If this completes successfully, is now accepted as proof of message authenticity.

The transmission of the public key with the encrypted message is very unusual. Although encrypted with the private key authenticates the sender, the inclusion of the public key can result in a potential attack. The attacker would be able to use the public key to decrypt the hash and modify it so that it would not pass the authentication phase, however the time stamp comes into effect here and prevents this from working and as the private key is secret the attacker would not be able to re-encrypt the hash. An attacker would have to discover an alternate key pair that hashes to the mobile's home address or perform a reply within the small time allowed.

The CAM protocol does not assure the trustworthiness or reliability of a node, other protocols should be used to provide this. This protocol also does not protect against distributed denial of service attacks that overwhelm the recipient with binding updates. Protocols such as IKE are recommended in [16] as an alternative to combat this attack. Another attack where the attacker jams a mobile's binding update yet sends it a binding acknowledgement cutting the mobile from the home agent and correspondent. This should be protected with IPSEC or a variant of CAM. To keep the implementation lightweight and interoperable [16] recommends the use of 1024-bit RSA for signing and SHA-1 for the public key hash.

4.3.4 CAM-DH Protocol

This is an improvement to CAM and uses a combination the BAKE/2 protocol [15] with a digitally signed Diffie-Hellman key exchange [4]. In CAM-DH a public signature key generates a mobile node's home address. The use of cryptographically generated addresses (CGA) [11] avoids the need for X.509 certificates [36] or similar mechanisms that associate keys with addresses [16]. Session keys are the product of a negotiation from a deffie-hellman exponent signed by the mobile private signature key. BAKE/2 portion of the protocol provides protection against denial of service attacks - while the signature mechanism provides a higher level of security than would be available with BAKE/2 used on its own.

The protocol works in following way:

Step 1 - The correspondent node is contacted by the mobile node, and receives its home and care-of addresses.

MN ⟶ CN : HoA,CoA

Step 2

CN ⟶ MN(HoA) : r_h, j, g^y

$r_h = MAC_{KCN}$ (HoA/N_j/0)

Step 3

CN ⟶MN(CoA) : r_c, j

$r_c = MAC_{KCN}$ (CoA/N_j/1)

In these two steps the correspondent node sends two challenges, one to the care of address and one to the home address. The correspondent also sends the mobile a Diffie-Hellman exponent.

Step 4 - The mobile hashes the two received challenges together to form a key (K3), uses the key to compute a message authentication code on its public key and signed Diffie-Hellman parameter. The purpose of the MAC is to convey the risk of the message being a forgery is low enough that it should reallocate computational resources to checking the signature and calculating the Diffie-Hellman exponent g^{xy}. The session key is calculated using a Diffie-Hellman key agreement that authenticates binding updates.

MN ⟶CN : T_0,HoA,CoA, i,MAC_{KBU} (T_0/HoAjCoA/i), g^x, S_{PK}(TypeTag/g^x/HoA), PK,MAC_{K3} (…), j

$K_3 = h(r_h/r_c)$

$K_h = h(g^{xy}/r_h)$

$K_{BU} = h(K_h/r_c)$

Step 5 - A binding request containing a fresh challenge is sent by the correspondent when its binding cache entry is about to expire.

CN ⟶MN : r'_c, j'

Step 6 - The mobile computes a new key K'_{BU}, sends a binding update authenticated using this key. K'_{BU} is calculated by hashing the old value of K_h together with the new challenge.

MN ⟶CN : T_0,HoA,CoA,i,$MAC_{K'BU}$(T_0/HoA/CoA/i), j'

$K_h = h(g^{xy}/r_h)$

$K'_{BU} = h\ (K_h/r'_c)$

All of the asymmetric cryptographic operations that the mobile carries out can be delegated to the home agent, provided that the home agent is given access to the appropriate keys. If the correspondent node is mobile, all of the asymmetric cryptographic operations that the correspondent performs can instead be performed by the correspondent's home agent.

4.3.5 Binding Update Backhauling

The Binding Update Backhauling (BUB) is a new mode designed to be used between two mobile endpoints. It is designed as an enhancement to the route optimization mode since it uses the same direct

path between the MN and CN for the data traffic exchange. The main objectives of the BUB mode are to reduce the number of mobility signaling messages exchanged between the two MNs and increase the security of what will remain. This is achieved by eliminating the HoTI/HoT and CoTI/CoT messages, diverting the BU messages to a more secure and reliable path going through the two HAs and keep using the direct path for the data traffic exchange [41].

The design of the proposed solution is based on the following:

a) The paths between the MNs and their HAs are protected by an ESP tunnel.

b) The path between the two HAs, being part of the backbone, is assumed to be more secure and more stable than the dynamic path between the two MNs.

c) A malicious node cannot be at the same time near the MNs and near the path going between the two HAs.

When both end points are mobile, the following four nodes, in Figure 9, become involved:

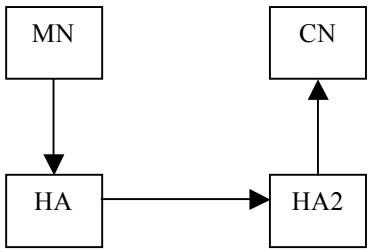

Figure 9. Binding Update Backhauling [41]

Switching to BUB mode should not occur before running a successful BUB test. The BUB test makes both endpoints agree on using the link going through the two HAs for exchanging the BU messages and MUST be run in parallel with the RR test.

It is true that sending binding updates via the home agents provides a substantial security advantage, however on closer inspection it is logical to assume that exchanging binding updates and other messages via the home agents will create an intolerable latency, especially in scenarios where both mobile nodes are great distances away from there home agents and the distance between home agent is also far.

4.4 Identity Protection

4.4.1 BLIND

BLIND is a security framework that provides identity protection against active and passive attacks for end-points. A two-round-trip authenticated Diffe-Hellman Key Exchange Protocol that protects the initiator's and responder's identity is presented in [42]. The protocol hides the public key based identifiers from attackers and eavesdroppers by blinding the identifiers. The protocol completes the identity protection by offering location privacy with forwarding agents. An end-point must negotiate a key exchange with its peer via the forwarding agent to obtain location privacy. The forwarding agent provides location privacy by hiding the real location of the node. The peers are able to see only the virtual address, not the real address of the end-point. A cryptographic hash of the public key end point identifier (EID) is called a fingerprint. Each party creates scrambled versions of the fingerprints and use each scrambled value only during one protocol run. This makes it impossible to correlate independent protocol runs.

4.4.2 Authorised Anonymous ID

To address the issue of location privacy, [43] introduces the idea of an authorised anonymous ID based scheme, which eliminates the need for a trusted server or administration. A cryptographic technique called blind signatures are used to generate an authorised anonymous ID which is used to replay the real ID of the mobile device. To address location privacy issues, an architecture was designed on the Wireless Andrew 802.11 WLAN network which used a centralized location server which stored the location data of registered mobile users. It is suggested in [43] that a distributed architecture would be more appropriate as a centralized architecture has drawbacks.

4.4.3 Temporal Mobile Identifier (TMI)

Various ways are suggested in [44] in which to prevent location information leakage. One way to do this is to hide the home address of the mobile node from third parties by using a temporal mobile identifier.

In MIPv6 packets transmitted contain the addresses of the mobile node and home address in clear text in the header. This can allow an eavesdropper to identify packets and track mobile movement. One solution is to use a Temporal Mobile Identifier (TMI) for each mobile node. This is a random 128 bit sequence which can identify the mobile node to other nodes. The TMI replaces the home address in the header of packets and has the effect of hiding the mobile home network identity from the correspondent and eavesdroppers.

An alternative method would be not to use binding updates at all and use bi-directional tunnelling. This means the correspondent sends all packets to the home address, which then encapsulates them and forwards them to the care of address.

If route optimisation is used then the binding update must contain the TMI in the home address option and the binding update must be encrypted.

4.4.4 Hierarchical Mobile IPv6

The hierarchical mobile IP management model [44] utilizes a new node called a mobility anchor point (MAP). It provides a central point to assist with hand offs. It can be located at any level in a hierarchical network including the access router (AR).

In the basic mode of Hierarchical mobile IP, the mobile node has two address, a regional care of address (RCoA) and on the MAP's subnet an on link care of address (LCoA). The MAP acts as a local home agent that maps the mobile node's regional care of address to its on link care of address. The mobile node has the option of hiding its on link care of address from the corresponding nodes and its home agent by using its regional care of address in the source field in the packets it sends. However an eavesdropper can still determine the mobile nodes home address by snooping the packets.

4.5 Other Technologies

4.5.1 Mobile Agents

Traditionally programs are executed on one machine; perform a task and end execution on the same machine. The next step in evolution for software is to become mobile. Tasks that have started execution on one machine can now be paused, "jump" to another computer and continue execution there. This is possible with mobile agents and opens up a new dimension in computer programming and usage.

Mobile agents are autonomous applications, which features the behavior of autonomy, social ability, learning and most importantly, mobility. Mobile agents can move from host to host in a heterogeneous network by saving its current state, performing a move to another host via data duplication and then resuming execution from the saved state. This means that they can control their own actions and move to different machines and execute on them at any time regardless of operation or operating system [45].

The traditional client/server model shows that the client sends a message to the server and the server replies (Figure 10).

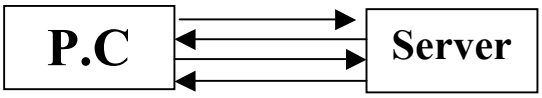

Figure 10. Client Server Model

They perform a continuous dialogue until the task is complete. Mobile agents work in a different way [46]. Their approach is to contain the user's data and instructions within the agent and dispatch it to a destination computer and there the agent communicates with the server at the server side (Figure 11).

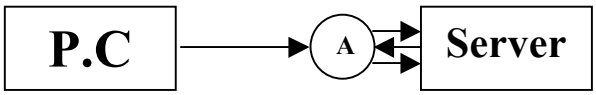

Figure 11. Agent communication with server

The benefit of this is that it reduces the network load and frees up bandwidth, it also allows for faster communication [47] [48].

This is a very attractive technology for the purposes of Mobile IP and as you will see in the proposed solution mobile agents can be used to facilities network messages and location privacy.

4.5.2 Long Term Evolution (LTE)

Long Term Evolution (LTE), LTE is a set of enhancements to the Universal Mobile Telecommunications System (UMTS), which addresses upgrading 3rd generation (3G) UMTS to 4th generation (4G) mobile communications technology to provide mobile broadband capabilities with enhanced multimedia services [49]. LTE is not a 4th generation (4G) technology, as it does not fully comply with 4G requirements. The pre-4G standard is a step toward LTE Advanced, a 4G standard of radio technologies designed to increase the capacity and speed of mobile telephone networks [50].

The advantages of the LTE specification are that it provides downlink rates of 100 Mbps, an uplink of 50 Mbps, high throughput, low latency, plug and play, frequency division duplexing (FDD) and time division duplexing (TDD) in the same platform and operates within an all-IP flat networking architecture. The air interface of LTE is E-UTRAN, which allows for even higher download speeds of 292 Mbit/s, however the disadvantage of LTE is that it is not backwards compatible with 3G systems [51].

The uptake of LTE technology is occurring at an incredibly fast pace. Just looking at announcements made by a leading U.S mobile operator over the course of a few months demonstrates this:

- September 2010, Verizon Wireless announced plans to have all their 3G customers switched over to 4G, so they can take advantage of the faster speeds, by the end of 2013 [52].

- January 2011, Verizon Wireless announced that it plans to add more than 100 LTE markets to cover 175 million people by the end of 2011, have 2/3 of the US population covered by mid-2012, and provide nationwide coverage in the U.S by the end of 2013 [53].
- March 2011, Verizon Wireless promised LTE in 147 markets by the end of 2011 [54].

The next step for LTE evolution is LTE Advanced and is currently being standardized.

4.5.3 Long Term Evolution Advanced (LTE Advanced)

LTE Advanced has been created by the 3rd Generation Partnership Project (3GPP) as a major enhancement of the 3GPP Long Term Evolution (LTE) standard [55]. LTE does not meet the International Telecommunication Union (IMT) requirements for the 4G specification, which defines the peak data rate of up to 1 Gbit/s. It is therefore considered a 3.9G technology.
WiMAX 2 has been approved for the LTE Advanced standard. It is designed to be backward compatible with WiMAX 1/1.5 devices and support is provided for conversion of earlier 'pre-4G' equipment. This allows LTE-Advanced to provide almost 3.3Gbit peak download rates per under ideal conditions [56]. LTE Advanced is backwards compatible with LTE and uses the same frequency bands. An LTE terminal should be able to work in an LTE-Advanced network and vice versa.

4.5.4 Dual Stack Mobile IPv6 (DS-MIPv6)

The main network architecture of Long Term Evolution (LTE) is Dual Stack Mobile IPv6 (DS-MIPv6) [51]. The DSMIPv6 specification extends the Mobile IPv6 capabilities to allow mobile nodes to forward IPv4 and IPv6 packets within the network. DSMIPv6 eliminates the need to run both IPv6 and IPv4 mobility management protocols simultaneously. As the majority of Internet devices are currently IPv4 based, the protocol gives the ability to nodes to tunnel IPv4 packets over IPv6, using a modified header [57]. IPv6 data can also be tunneled over IPv4 meaning that mobile nodes need only MIPv6 to manage mobility while moving within both IPv4 and IPv6 networks. Network Address Translator (NAT) Traversal support is also provided to routers using DSMIPv6 [58].

4.6 Discussions

The Mobile IP working group is currently searching for a security solution that enables semi-secure, weak authentication between IPv6 Mobile Nodes and correspondent Nodes in the global Internet.
A less than perfect security solution is necessary in this situation because strong authentication between previously unknown peers would require a global Public Key Infrastructure (PKI). This is neither possible nor desirable with the current Mobile IPv6 infrastructure. The purpose of the weak authentication mechanism is to establish a Binding Security Association (BSA) between the MN and the CN for the secure exchange of Binding Updates (BUs). There are various alternatives on how the

binding updates (BUs) can be protected, but most of them fall under the category of "Use IPSec for everything" or "Don't Use IPSec At All" [59].

Strong authentication may be offered as an alternative to weak authentication for certain networks or can be used simultaneously with the weak methods. Typical security technologies allow users to define on which IP addresses or networks certain methods should be used. However this opens an attack where the source address is modified to claim no security is needed because the packet source address has been forged to reside within the secure network [60]. This can fool the destination if it relies on addresses in its security policies.

The cost of strong authentication may exceed the benefits. Mobile IPv6 does not justify the introduction of global public key infrastructure for the sole purposes of authenticating nodes participating in the optimisations. In some cases the use of weak authentication, makes the cost of the attack to the attacker exceed the value of the data [61]. However binding updates are very valuable, but perhaps the use of a timestamp in conjunction with weak authentication may give the necessary delay that would render the information gained useless in the case of replay attacks.

The use of public key cryptography is essential as it is an effective way of determination the authenticity of the data especially in the case of digital signatures [4]. However the high cost of using PKI such as RSA in mobile devices is too steep for the technology to efficiently utilize. However the implementation of elliptic curve cryptography [21] can reduce the cost to resources without compromising its strength and effectiveness.

However it is possible to have an authentication protocol, which does not use public key cryptography such as the symmetric key protocol. This is the simplest of the binding update protocols and is not very resource intensive however the problem arises of how to distribute the keys with out them being intercepted.

Optimisations to the symmetric key protocol can allow it to be used with manually configured shared secret keys between the mobile and home agent where the agent maintains a database of issued keys. Another idea is to use a certificate based shared secret key agreement can be used to associate a node's public key with its home address allowing the PKI infrastructure to authenticate the home address.

the shared key protocol can be extended dynamically establish the shared secret The BAKE/2 protocol [15]. However communications between nodes must be protected from eavesdropping by security not supplied by this protocol, such as IPSEC [12]. Another drawback is that the protocol does not defend against an attacker who can monitor the home agent to correspondent node route. It can however protect the correspondent node against denial of service attacks, which flood it with bogus messages. This prevents resource exhaustion because large amounts of processing power are not used to handle

yet to be authenticated messages. This protocol is suitable for communication between a mobile node and a non-mobile server but may not be suitable for communication with a mobile server.

The BAKE/2 protocol protects against dos attacks where the attacker uses the victims care of address to redirect high bandwidth traffic to it. Flow control protocols such as TCP do not defend against this attack because the acknowledgements can be forged. The protocol succeeds because it only completes with participants of it.

CAM-DH uses a combination the BAKE/2 protocol [15] with a digitally signed Diffie-Hellman key exchange. This protocol can be optimised if all of the asymmetric cryptographic operations that the mobile carries out can be delegated to the home agent, provided that the home agent is given access to the appropriate keys. Another optimisation can occur if the correspondent node is mobile, then all of the asymmetric cryptographic operations that the correspondent performs can instead be performed by the correspondent's home agent.

The BAKE/2 and CAM-DH protocols prevent dos attacks by verifying that packets sent to a mobile's claimed care of address reach a willing participant of the protocol preventing redirection attacks. These protocols also do not authenticate the care of address. If an attacker intercepts packets sent to the care of address then it will be able to complete the protocol and flood the unwilling care of address with data. Deriving the care of address and the home address from the nodes public key is alternative method of authenticating them however this was not used in BAKE/2 or CAM-DH because of restrictions imposed on them by the sub-networks.

The one thing that stands out of all of these security protocols is that in their very first message, they transmit the home address and care of address to the correspondent in plain sight. This gives away their location, which can be the bases for a number of attacks. A solution must be found to initiate a binding update without giving this confidential information away to potential attackers.

To address the issue of location privacy, [43] introduces the idea of an authorised –anonymous ID based scheme, which eliminates the need for a trusted server or administration. A cryptographic technique called blind signatures is used to generate an authorised anonymous ID, which is used to replay the real ID of the mobile device. To address location privacy issues, an architecture was designed on the Wireless Andrew 802.11 WLAN network which used a centralized location server which stored the location data of registered mobile users. It is suggested in [43] that a distributed architecture would be more appropriate, as a centralized architecture has drawbacks which include, location privacy of mobile users is not under the users control, central server is a single failure point, and a successful attack would compromise location privacy, and that a centralized architecture is not scalable. However this system requires initial authentication from an infrastructure that supports either the public key (PKI) or a Kerberos based system.

The work is not designed for mobile IP but it is related as mobile IP shares a similar structure. However they are essentially different in two ways. Mobile IP is concerned with packet forwarding and routing and this system is about providing a location service to the user. The second difference is that they operate at different layers as Mobile IP works at the network layer and this system works at the application layer.

4.7 Summary

Numerous security solutions have been proposed and each have their advantages and disadvantages. The two main types of security are encryption and authentication. Encryption protects the confidentiality of the data and comes in two flavours, symmetric and asymmetric. The former is useful for low powered devices and participants use the same key to encrypt and decrypt. The problem is how to distribute the key without it being intercepted. Asymmetric keys are split in to encryption and decryption keys. This is useful for the distribution of the keys and can help with authentication with the use of digital signatures. The drawback however is that processing consumption is 100 – 1000 times that of symmetric cryptography. This can be reduced somewhat with the implementation of elliptic curve cryptography which is a lightweight public key cryptographic solution.
Authentication allows users to verify that they are communicating with validated participants. Different systems exist, such as Kerberos that perform authentication by referring to a central authentication database to compare users credentials. However in the mobile IP architecture it is best to stay away from centralized authorities as they are a single entity and hence a single point of attack. A distributed authentication system is required which use techniques such as hashes, digital signatures, address based keys and cryptographically generated addresses. Address based keys and certified addresses could be used for signing address resolution, duplicate address detection and redirection messages however, cryptographically generated addresses have the advantage that no trusted third parties are required
More elaborate systems such as RADIUS based on AAA Authentication, authorization and accounting, allow for a combination of security but must rely on an authentication server.
Currently it is recommended that all mobile IP security be handled by IPSEC. However the cost in resources to utilize IPSEC is beyond what it realistically expected from a mobile device in effect reducing the users quality of service.

Security protocols have been specifically designed for the protection of binding updates such as, Bake/2 and CAM, from eavesdropping modification and DOS attacks. However they all make the same fundamental error. They give away the location of the home agent, the mobile node and the correspondent. This is the basis for many of the possible attacks to these nodes.

A solution must be developed that allows for the crucial location information to be transmitted and yet the nodes retain their location privacy. Systems have been developed that do this at some level such as the hierarchical mobile IP management model's use of the mobility anchor point (MAP). However location privacy is moot point with the introduction of cryptographically generated addresses. This

allows users to assert their ownership over an address preventing spoofing. This combined with return routability may provide secure solution.

What is needed is a system can be developed that can fulfil the security needs of mobile IP's vulnerabilities by using a combination of the security technologies available, which operate without over taxing the computing resources available and package them into an easy to implement solution.

Chapter 5 Proposed Solution

5.1 Protocol Design Considerations

To design a security solution for Mobile IP binding updates, first the security vulnerabilities must be known. These are presented in chapter 3. There are a variety of vulnerabilities but many of them are a result of the attacker knowing location information of the communicating nodes. This creates an intriguing paradox, how can you transmit your location data without giving your address away, or at least to an unauthorised party. A logical answer would be to encrypt the data. But to send encrypted data, a secret key must first be exchanged and for nodes to exchange keys they must know each other's addresses. And so the vicious circle goes on. However it may be possible to use a third party intermediary to help facilitate authentication and key establishment without revealing the location of the communicating parties until they have been authenticated. Similar schemes have been employed to hide the location of the mobile node with Hierarchical Mobile IP and BLIND discussed in chapter four. The Trinity protocol [62] attempted to solve this problem by introducing a Binding Update Agent but by doing so created more problems for its primary goal of authentication. The issue was that while providing location privacy to the Mobile Node it introduced new elements into the infrastructure which themselves were vulnerable to impersonation and man in the middle attacks. However this solution was modified which later became the security solution of Mobile Home Agents.

Chapter four also discusses various security protocols available, each with their advantages and disadvantages. The BAKE/2 and CAM–DH are effective binding update protocols but suffer some weaknesses. They are an ideal starting point for building a security protocol. An enhancement to the protocol should be the concealment of the mobiles nodes address until authentication has taken place. Asymmetric cryptographic should be chosen over symmetric, for the secure transmission of the binding update data, as it affords more security options such as digital signatures, which can be used to authenticate user data. Asymmetric cryptography is more processor intensive then symmetric and so a light weight algorithm should be chosen to compensate for this such as Elliptic curve.

If a third party node is used then secondary security can be implemented to aid in the prevention of DOS attacks. This can be done with some form of weak authentication and ingress filtering effectively turning the node in to a firewall. Unfortunately ingress filtering must be applied to the attacker's local network, which is out of the control of any potential victim

Additions to the binding update protocol should include time stamps, to guard against replay and spoofing attacks, hashes for data integrity checks and perhaps even the use of Cryptographically Generated Addresses to bind the address to the user's key. Return Routability should also be used to verify that the node is in the location that it claims to be. Combining Cryptographically Generated Addresses and Return Routability provides a strong and complimentary solution.

However several considerations have been taken in to account such as network latency caused by the addition of a third party node and also the delay caused by the processing cost of the algorithm. If the algorithm were very secure but too slow to be realistically utilised then the protocol would have failed, likewise if it is fast but provides no security. A balance must be reached, which will be determined during simulations.

The proposed protocol removes any need for third party nodes for the simple reason that their implementation would incur extra cost, which the industry would be reluctant and unlikely to pay for. Therefore the proposed solution is designed with the following three premises:

1. Use only technology, which is already available without changing the current Mobile IPv6 architecture.
2. Design and focus on an effective authentication architecture.
3. Design the security solution so as it is not overwhelmingly processor intensive to mobile devices.

Trinity [62] attempted to be an all in one solution, however the introduction of new elements, such as the binding update node, into the Mobile IPv6 infrastructure overcomplicated its purpose of creating a secure and optimised binding update procedure. The addition of new nodes also introduced new security threats, which defeated the purpose of the solution, as it did not work with an unmodified network. It also would not adapt to any changing developments in Mobile IPv6 architecture. The new proposed solution will concentrate on authentication and attempt to keep it uncomplicated and within the above three listed goals.

5.2 The Protocol

The proposed solution attempts to improve the security of binding updates by adding an extra level of authentication. Most authentication systems operate on the premise of a third party is added to the system to provide the authentication requires. In mobile IPv6 for telecommunication devices, however, this is unnecessary. We assume that every mobile node needs to subscribe to a network provider, which in turn will provide the user with a home agent. This is the initial point of contact when an entity wishes to communicate with the mobile device as the home agent is constantly tracking and monitoring its current location.

Using today's mobile communications as a template we know that current systems use a sim card, which contains a sim number and the phone number. The device in use also has an IMEI, which is the hardware serial number. All of these are registered with the service provider. This information will be the basis for the new security solution.

5.2.1 Distributed Authentication Protocol

There are three main aspects to the security protocol:

1. Cryptographically Generated Addresses
2. Return Routability
3. Authentication verification

The first two technologies are well-established techniques. The third has been modified specifically for the protection of binding updates. Cryptographically Generated Addresses provide a reasonable assurance that the address of the user is indeed owned by them and not spoofed. Return Routability provides location authentication proving the communicating device is at the IP address claimed and again combats spoofing.

The third aspect of the security protocol provides device authentication and can be expanded to include user authentication in case of device theft.

Adding security features means that there will be an increase in processing power needed by devices. To aid with this burden the protocol proposes using a distributed authentication architecture, which can use other nodes such as the Home Agent to aide with processing tasks. The Correspondent Node requests authentication data from the Home Agent and the Mobile Node. The Home agent stores the data as a hash, which is unreadable by any attacker who would try to intercept it when transmitted. The Mobile Node sends the plane text data, which of course could be intercepted. To protect the plain text data on its way to the correspondent, the Mobile Node can encrypt it with the binding key created from the Return Routability stage.

Both pieces of authentication data are sent to the correspondent where the encrypted data from the Mobile is deciphered and then the data is hashed. The two strings are then compared and if they match the authentication process passes.

The Distributed Authentication Protocol provides a decentralised authentication system when there is no central authority. Each Mobile Nodes' security data, such as its Care of Address, Sim No, IMEI, Phone No. and Biometric data, is stored with its own Home Agent which is maintained by the Internet Service Provider which manages it. This provides a safe and secure authentication infrastructure without any one single point of attack. If a Home Agent is attacked it will not effect anyone else using the system as each Mobile Node has it's own Home Agent.

This protocol is designed to be used with either two communicating mobile nodes or a mobile node communicating with a static correspondent. In either case two options are presented:

1. Distributed authentication
2. Standard authentication.

The first protocol to be looked at is:

5.2.1.1 Standard and Distributed Authentication in Mobile-to-Mobile Communication.

Firstly all nodes should be using cryptographically generated address, which have been previously created by the function discussed in [18]:

Host ID = $HASH_{62}$(public key)

A cryptographic hash of the public key is created, called the Host ID, and this is used as the interface identifier part of the Mobile IPv6 address. The first 64 bits are the network prefix and are unchanged, however the last 64 bits of the address are now bound to the public key. The node can then claim ownership of the address by reversing the procedure, which can be done with a conventional public key signature. Now that an address has been created, which the Mobile Node can prove ownership of, the protocol sends it's first message.

Message 1.
The mobile node MN attempts to contact the mobile correspondent node CN. It does not know its current location so begins the Return Routability procedure contacts the correspondent's home agent HA2. The mobile node's care of address CoA and home address HoA are signed using it's private key MNK- and is sent with the Mobile Node's public key, MNK+, to HA2 the correspondent's home agent.

MN ⟶ HA2: MNK+, MNK-(CoA, HoA).

Message 2.
The correspondent's home agent forwards the data to the correspondent by using IP within IP to encapsulate the packets. This is done by adding another packet header to the data with the destination address of the Corresponding node.

HA2 ⟶ CN: MNK+, MNK-(CoA, HoA).

Message 3.
The corresponding node uses the public key MNK+ to decrypt the CoA and HoA addresses which proves that the sender is the only one who can encrypt and sign the messages. The correspondent then compares the mobile node's public key MNK+ with that of its claimed CGA address, CoA, and determines if they match. This is done by hashing the public key of the Mobile Node and comparing it with the interface identifier of the Mobile Nodes' claimed address. If they match then address

ownership has been proven. Once proven then Return Routability and device authentication will proceed, otherwise the connection / binding update request is denied.

The next step the correspondent will perform the home address check and the care of address check. The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key K_{cn} only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

Home token = hash (K_{cn} | source address | N_H | 0)

This is then sent to the home agent.

CN ⟶ HA: HoT.

Message 4.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT.

Message 5.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 3, however the token generated is slightly different. The nonce of the Care-of token is different from that of the Home Token and is denoted as N_C. The nonces are randomly generated bit strings that are changed periodically. The correspondent node keeps track of them using indices.

Care-of token = hash (K_{cn} | source address | N_C | 1)

This is then sent directly to the mobile node within a Care of test (CoT) packet.

CN ⟶ MN: CoT.

Message 6.

The mobile node receives both tokens from both the test packets sent. It then creates a binding key K_{bm} by hashing the two tokens together.

K_{bm} = hash (home token | care-of token)

The key is used to protect the first and following binding updates by encrypting them with the secure key. The mobile node then sends a binding update request to the correspondent node, which is

protected with the binding key $K_{bm.}$ As the Correspondent is one who sent the tokens, it is the only node capable of decrypting the binding update.

MN ⟶ HA2: K_{bm}(BU)

Message 7.

Notice that mobile node still sends its packets via the correspondent nodes home address. This is because both nodes are mobile and the MN would have to accept a binding update from the correspondent before being able to communicate directly. For now the correspondents home agent HA2 forwards the packet to the correspondent.

HA2 ⟶ CN: K_{bm}(BU)

Message 8.

This is where traditionally the correspondent would decrypt the data and accept the binding update, however before this begins it must wait for the result of the distributed authentication protocol to complete. This is to avoid possible denial of service attacks with repeated decryption requests. This authentication takes place simultaneously with return routability.

The correspondent node sends a request message to the mobile node for its authentication data (RAD).

CN ⟶ MN: RAD

Message 9.

The mobile node replies to the message by sending its authentication data, which includes its current address, its sim number, IMEI number, phone number and even and options for user authentication such as biometric data and a Time Stamp. This sent to the CN via HA2 encrypted with the binding key $K_{bm.}$

MN ⟶ HA2: K_{bm}(CoA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 10.

HA2 ⟶ CN: K_{bm}(CoA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 11.

Simultaneously to message 8, the correspondent sends a request for authentication data message to the home agent.

CN ⟶ HA: RAD

Message 12.

The home agent does not have the binding key so sending the authentication data would be a security risk. Instead the home agent hashes the authentication data together and sends that to the correspondent.

HA ⟶ HA2: Hash(CoA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 13.

HA2 ⟶ CN: Hash(CoA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 14.

Now the correspondent will have both the hash of the authentication data and the authentication data encrypted with the binding key. There are now two options:

1. The correspondent performs the authentication comparison by decrypting the binding key and hashing the authentication data received from the mobile node then comparing this to the hash received by the home agent. Skip to Message 16.
2. If the correspondent node is overwhelmed by the current processing constraints it may opt to send the hash and the decrypted authentication data to the correspondents home agent via a secure tunnel where it will perform the comparison. (Notice the key is not sent as this would be a security vulnerability. The decryption is done by the CN.)

CN ⟶ HA2: (Hash(CoA, Sim No, IMEI, Phone No., Biometric), (CoA, Sim No, IMEI, Phone No., Biometric, Timestamp))

Message 15.

The HA2 hashes the authentication data and compares it to the hash. If they match then the authentication is successful and an authorisation ok message is sent to the correspondent node.

HA2 ⟶ CN: AOK

Message 16.

If the result of the authentication is successful then the binding update received in message 7, $K_{bm}(BU)$ is decrypted using the Correspondents shared key as the protocol uses symmetric key cryptography only known to the Mobile and Correspondent Nodes. The binding update is accepted and a binding acknowledgement BA is sent to the mobile node allowing it to communicate directly with the correspondent.

CN ⟶ MN: BA

As the correspondent is also mobile, the mobile node will have to accept binding updates from it also. The process is the same only in reverse. Of course less messages will be needed as the mobile node can now communicate directly with the correspondent unless it takes place at the same time. All message exchanges can be seen in Figure 12.

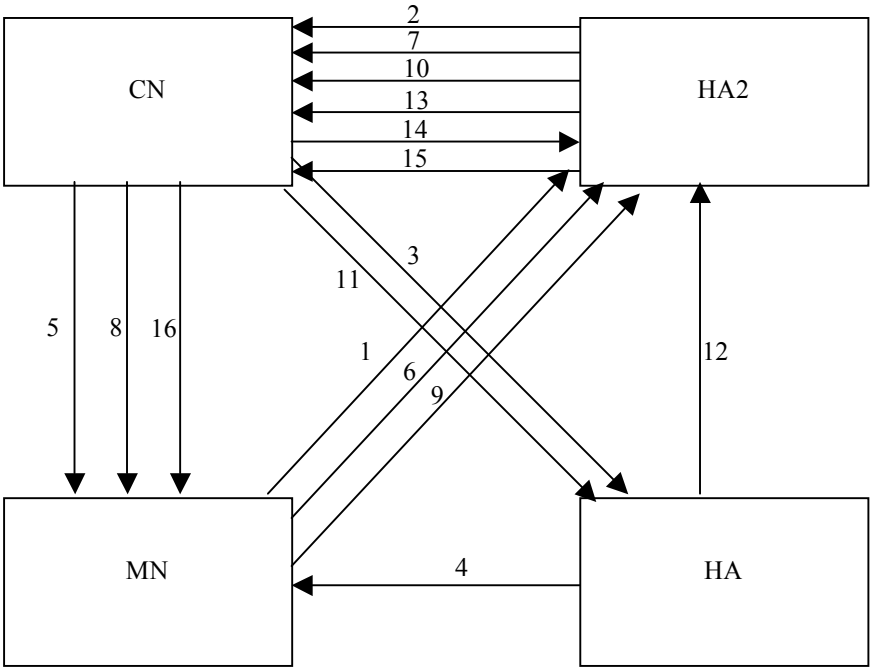

Figure 12. Home agent distributed authentication.

5.2.1.2 Authentication in Mobile-to-Static Communication.

The principle is the same as the mobile-to-mobile communication however as the correspondent is static it does not have a home agent and so cannot perform distributed authentication. However as it is not mobile it is logical to assume it will have a great deal more processing power than a mobile device and so distributed authentication would be unnecessary. All the messages are the same but are far fewer as they do not pass via the correspondent's home agent. All messages can be seen in Figure 13.

Message 1.

The mobile node MN attempts to contact the correspondent node CN. The mobile node's care of address CoA and home address HoA are signed using it's private key MNK- and is sent with the Mobile Node's public key, MNK+, to the correspondent.

MN ⟶ CN: MNK+, MNK-(CoA, HoA).

Message 2.

The corresponding node uses the public key MNK+ to decrypt the CoA and HoA addresses which proves that the sender is the only one who can encrypt and sign the messages. The correspondent then compares the mobile node's public key with that of its claimed CGA address and determines if they match. If they do then return routability and device authentication will proceed, otherwise the connection / binding update request is denied.

The next step the correspondent will perform the home address check and the care of address check.
The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key K_{cn} only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

Home token = hash (K_{cn} | source address | N_H | 0)

This is then sent to the home agent.

CN ⟶ HA: HoT.

Message 3.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT.

Message 4.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 2, however the token generated is slightly different. The nonce of the Care-of token is different from that of the Home Token and is denoted as N_C. The nonces are randomly generated bit strings that are changed periodically. The correspondent node keeps track of them using indices.

Care-of token = hash (K_{cn} | source address | N_C | 1)

This is then sent directly to the mobile node within a Care of test (CoT) packet.

CN ⟶ MN: CoT.

Message 5.

The mobile node receives both tokens from both the test packets sent. It then creates a binding key K_{bm} by hashing the two tokens together.

K_{bm} = hash (home token | care-of token)

The key is used to protect the first and following binding updates. The mobile node then sends a binding update request to the correspondent node, which is protected with the binding key $K_{bm.}$

MN ⟶ CN: K_{bm}(BU)

Message 6.

This is where traditionally the correspondent would decrypt the data and accept the binding update, however before this begins it must wait for the result of another authentication protocol to complete. This authentication takes place simultaneously with return routability.

The correspondent node sends a request message to the mobile node for its authentication data (RAD).

CN ⟶ MN: RAD

Message 7.

The mobile node replys to the message by sending its authentication data, which includes its current address, its sim number, IMEI number, phone number and even and options for user authentication such as biometric data and timestamps. This sent to the CN encrypted with the binding key $K_{bm.}$

MN ⟶ CN: K_{bm}(CoA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 8.

Simultaneously to message 8, the correspondent sends a request for authentication data message to the home agent.

CN ⟶ HA: RAD

Message 9.

The home agent does not have the binding key so sending the authentication data would be a security risk. Instead the home agent hashes the authentication data together and sends that to the correspondent.

HA ⟶ CN: Hash(CoA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 10.

Now the correspondent will have both the hash of the authentication data and the authentication data encrypted with the binding key.

The correspondent performs the authentication comparison by decrypting the binding key and hashing the authentication data received from the mobile node then comparing this to the hash received by the home agent.

If the result of the authentication is successful then the binding update is accepted and a binding acknowledgement BA is sent to the mobile node allowing it to communicate directly with the correspondent.

CN ⟶ MN: BA

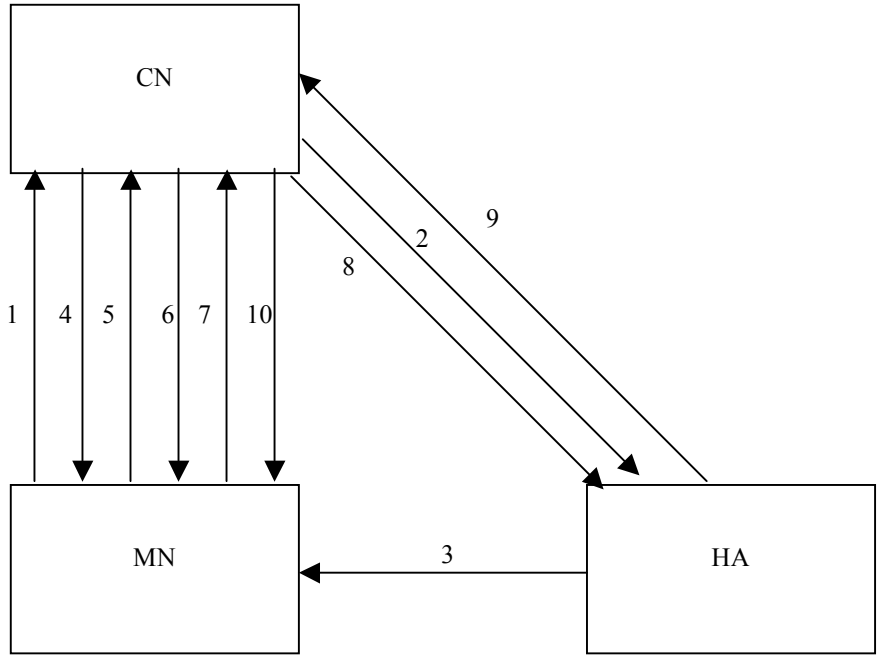

Figure 13. Authentication in mobile to static node communication.

The security protocol uses the same mechanisms, which are already in place, meaning the architecture and headers, Figure 14, remain the same insuring compatibility with any future mobile IP system.

Ver.	Traffic Class	Flow label
Payload Length	Next header	Hop limit
128 Bit Source Address		
128 Bit Destination Address		

Figure 14. IPv6 header

5.2.1.3 Summary

The protocol is made up of three components. Cryptographically Generated addresses, to assert ownership of the IP address the node claims to reside on. Return Routability, a solution to determine the node is at the location it claims to be and is reachable. Both these solutions currently exist. The third element proves the owner of the node is authorised to communicate across the network using a distributed authentication mechanism where authentication data from the ISP is compared to that stored on the home agent and mobile node which provides low resource consumption solution.

The advantages of using a distributed authentication protocol is that there is a predicted increase in processing speed concerning the completion of security techniques which at the same time not over burdening the mobile processor with all the work.
The disadvantage is that there is an increase in network traffic, however optimisation to the protocol may be able to reduce this.

5.2.2 Dual Identity Return Routability

In addition to the distributed authentication protocol, a new solution in now proposed as a modification to return routability. It will demonstrate that dual identity phones can be used to improve security within 4G networks.

Dual Identity Return Routability provides an extra layer of security to the standard Return Routability solution by taking advantage of a device, which utilises a phone with two sim cards or a single dual identity sim. As each identity has a separate connection and IP address, tokens are sent to both identities. This prevents spoofing as only the user, which has both identities in the same device will be able to received the tokens and create the binding key. This can only be achieved by having the unique mobile hardware provided by the dual identity sims and the appropriate device operating system to utilise it.

Dual identity return routability is part of a larger security solution, but could be used as a stand-alone solution. Before the protocol takes place the mobile node sends an intention to communicate with the correspondent node. The mobile node sends the correspondent node its public key MNK+, care of address CoA (actual location dynamic address) and its home address HoA (static address) and the phone number address of its other identity CoA2 and its home agent address HoA2. It is possible for both identities to share the same home agent Figure 15, or use separate home agents Figure 16. This is different from the Return Routability section of the Distributed Authentication Protocol as the Mobile Node now sends the Correspondent two care of addresses and two home addresses, CoA1 and HoA1 belonging to one of the identities and CoA2 and HoA2 belonging to the other.

MN ⟶ CN: MNK+, CoA1, HoA1, CoA2, HoA2.

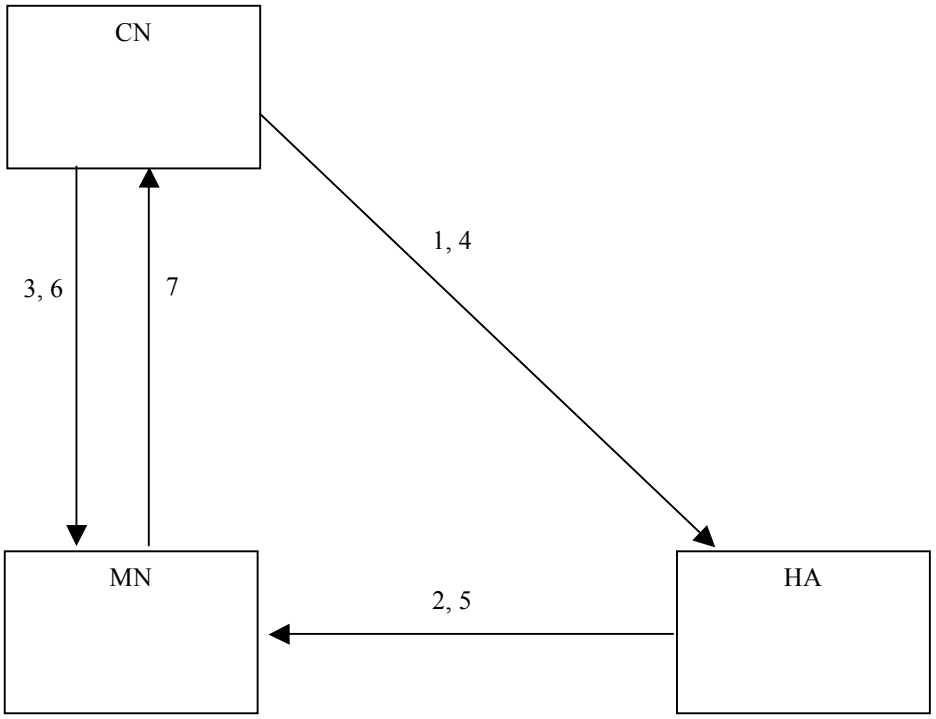

Figure 15. Dual Identity Return Routability with both identities sharing the same home agent.

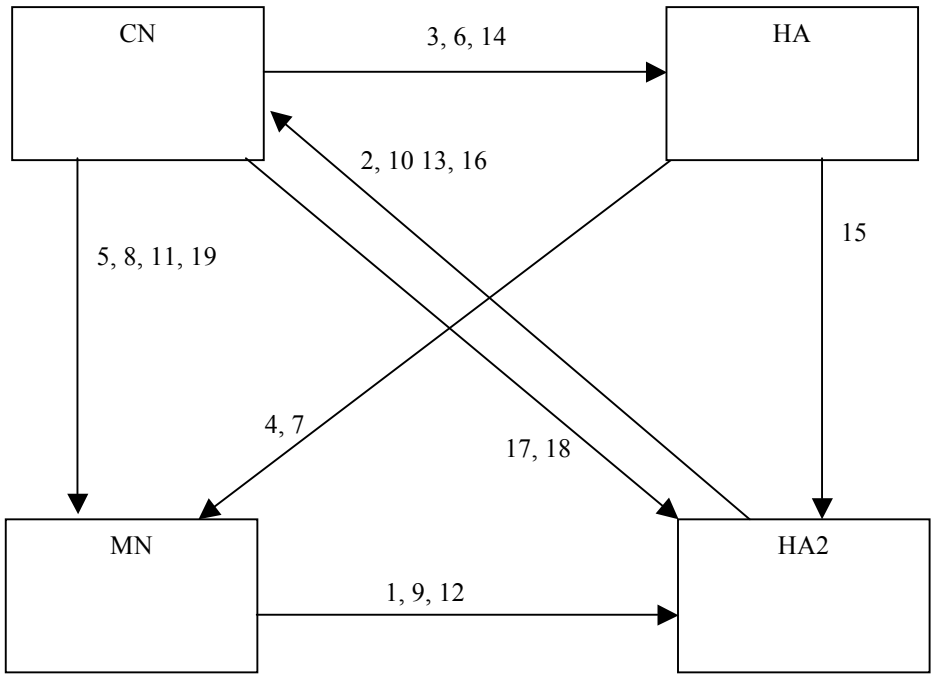

Figure 16. Dual Identity Return Routability with both identities using different home agents.

The Correspondent node (CN) will test to see if the Mobile node MN is reachable at the care of address and also test the other identity address is reachable. The two identities are linked together making spoofing a lot more difficult and proving the user is the owner of the identity.

5.2.2.1 Distributed Authentication Protocol with Dual Identity Return Routability in Mobile-to-Mobile Communication.

This section describes how dual identity return routability is incorporated in the updated distributed authentication protocol.
Firstly all nodes should be using cryptographically generated address, which have been previously created by the function discussed in [17]:

$$\text{Host ID} = \text{HASH}_{62}(\text{public key})$$

Message 1.
The mobile node MN attempts to contact the mobile correspondent node CN. It does not know its current location so it first contacts the correspondent's home agent HA2. The mobile node's care of address CoA and home address HoA are signed using it's private key MNK- and is sent with the Mobile Node's public key, MNK+, to HA2 the correspondent's home agent. Message flows are shown in Figure 16.

MN ⟶ HA2: MNK+, MNK-(CoA, HoA).

Message 2.
The correspondent's home agent forwards the data to the correspondent by encapsulating the packets.

HA2 ⟶ CN: MNK+, MNK-(CoA, HoA).

Message 3.
The corresponding node uses the public key MNK+ to decrypt the CoA and HoA addresses which proves that the sender is the only one who can encrypt and sign the messages. The correspondent then compares the mobile node's public key with that of its claimed CGA address and determines if they match. If they do then the new dual identity return routability and device authentication will proceed, otherwise the connection / binding update request is denied. The next step the correspondent will perform the home address check and the care of address check. The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key K_{cn} only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

Home token1 = hash (K_{cn} | source address | N_H | 0)

This is then sent to the home agent.

CN ⟶ HA: HoT1.

Message 4.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT1.

Message 5.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 1 and 2, however the token generated is slightly different. The nonce of the Care-of token is different from that of the Home Token and is denoted as N_C. The nonces are randomly generated bit strings that are changed periodically. The correspondent node keeps track of them using indices.

Care-of token1 = hash (K_{cn} | source address | N_C | 1)

This is then sent directly to the mobile node within a Care of test (CoT) packet.

CN ⟶ MN: CoT1.

Message 6.

Home token2 = hash (K_{cn} | source address | N_{H2} | 2)

This is then sent to the home agent of the second identity.

CN ⟶ HA: HoT2.

Message 7.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT2.

Message 8.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 4 and 5, again the token generated is slightly different and the nonces are also different with N_{H2} and N_{C2} for the second token pair.

Care-of token2 = hash (K_{cn} | source address | N_{C2} | 3)

This is then sent directly to the mobile node within a Care of test (CoT) packet.

CN ⟶ MN: CoT2.

Message 9.

The mobile node receives all four tokens from the four test packets sent. It then creates a binding key K_{bm} by hashing the four tokens together.

K_{bm} = hash (home token | care-of token | home token2 | care-of token2)

The key is used to protect the first and following binding updates. The mobile node then sends a binding update request to the correspondent node, which is protected with the binding key K_{bm}

MN ⟶ HA2: K_{bm}(BU)

Message 10.

The mobile node still sends its packets via the correspondent nodes home address. This is because both nodes are mobile and the MN would have to accept a binding update from the correspondent before being able to communicate directly. For now the correspondents home agent HA2 forwards the packet to the correspondent.

HA2 ⟶ CN: K_{bm}(BU)

Message 11.

This is where traditionally the correspondent would decrypt the data and accept the binding update, however before this begins it must wait for the result of another authentication protocol to complete. This authentication takes place simultaneously with dual identity return routability.
The correspondent node sends a request message to the mobile node for its authentication data (RAD).

CN ⟶ MN: RAD

Message 12.

The mobile node replies to the message by sending its authentication data, which includes its current address, its sim number, IMEI number, phone numbers/identities and even and options for user authentication such as biometric data and time stamps. This is sent to the CN via HA2 encrypted with the binding key $K_{bm.}$

MN ⟶ HA2: K_{bm}(CoA, Sim No, IMEI, Phone No. 1 + 2, Biometric, Timestamp)

Message 13.

HA2 ⟶ CN: K_{bm}(CoA, Sim No, IMEI, Phone No. 1 + 2, Biometric, Timestamp)

Message 14.

Simultaneously to message 11, the correspondent sends a request for authentication data message to the home agent.

CN ⟶ HA: RAD

Message 15.

The home agent does not have the binding key so sending the authentication data would be a security risk. Instead the home agent hashes the authentication data together and sends that to the correspondent.

HA ⟶ HA2: Hash(CoA, Sim No, IMEI, Phone No. 1 + 2, Biometric, Timestamp)

Message 16.

HA2 ⟶ CN: Hash(CoA, Sim No, IMEI, Phone No. 1 + 2, Biometric, Timestamp)

Message 17.

Now the correspondent will have both the hash of the authentication data and the authentication data encrypted with the binding key. There are now two options:

1. The correspondent performs the authentication comparison by decrypting the binding key and hashing the authentication data received from the mobile node then comparing this to the hash received by the home agent. Skip to Message 19.
2. If the correspondent node is overwhelmed by the current processing constraints it may opt to send the hash and the decrypted authentication data to the correspondents home agent via a secure tunnel where it will perform the comparison. (Notice the key is not sent as this would be a security vulnerability. The decryption is done by the CN.)

CN⟶HA2: (Hash(CoA, Sim No, IMEI, Phone No. 1 + 2, Biometric, Timestamp), (CoA, Sim No, IMEI, Phone No. 1 + 2, Biometric, Timestamp))

Message 18.

The HA2 hashes the authentication data and compares it to the hash. If they match then the authentication is successful and an authorisation ok message is sent to the correspondent node.

HA2 ⟶ CN: AOK

Message 19.

If the result of the authentication is successful then the binding update is accepted and a binding acknowledgement BA is sent to the mobile node allowing it to communicate directly with the correspondent.

CN ⟶ MN: BA

As the correspondent is also mobile, the mobile node will have to accept binding updates from it also. The process is the same, only in reverse. Of course less messages will be needed as the mobile node can now communicate directly with the correspondent unless it takes place at the same time.

5.2.2.2 Distributed Authentication Protocol with Dual Identity Return Routability in Mobile-to-Static Communication.

Message 1.

The correspondent will perform the home address check and the care of address check. The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key K_{cn} only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

Home token1 = hash (K_{cn} | source address | N_H | 0)

This is then sent to the home agent.

CN ⟶ HA: HoT1.

Message 2.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT1.

Message 3.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 1 and 2, however the token generated is slightly different.

Care-of token1 = hash (K_{cn} | source address | N_C | 1)

This is then sent directly to the mobile node within a Care of test (CoT) packet.

CN ⟶ MN: CoT1.

Message 4.

Home token2 = hash (K_{cn} | source address | N_{H2} | 2)

This is then sent to the home agent of the second identity.

CN ⟶ HA: HoT2.

Message 5.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT2.

Message 6.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 4 and 5, again the token generated is slightly different and the nonces are also different with N_{H2} and N_{C2} for the second token pair.

Care-of token2 = hash (K_{cn} | source address | N_{C2} | 3)

This is then sent directly to the mobile node within a Care of test (CoT) packet.

CN ⟶ MN: CoT2.

Message 7.

The mobile node receives all four tokens from the four test packets sent. It then creates a binding key K_{bm} by hashing the four tokens together.

K_{bm} = hash (home token | care-of token | home token2 | care-of token2)

The key is used to protect the first and following binding updates. The mobile node then sends a binding update request to the correspondent node, which is protected with the binding key K_{bm}

MN ⟶ CN: K_{bm}(BU)

This protocol proves that the mobile node is reachable at its current address, preventing denial of service attacks and proves that the two identities are associated with each other proving ownership of the phone numbers / IP addresses and providing a cheap method of authentication.

5.2.2.3 Summary

The research attempted to discover the different types of future technologies in development, which may be incorporated into the 4G fourth Generation mobile network. One technology stood out which allows multiple phone numbers, or identities, to be simultaneously used on a single sim card. Different wireless transmission technologies were also investigated such as W-FI and WiMAX.

Taking advantage of these technologies, a new security solution was created based on return routability. Secret tokens are sent to the addresses of the mobile node and the home agent of both identities. As they are sent via different paths and two of the four secret token are sent via an alternative network, it is highly unlikely that all of the tokens can be intercepted by an attacker and all of them would be needed to create the binding key. This provides reasonable reassurance of two things, 1, the mobile node is indeed in the location it claims to be, as it proves that the correspondent node can communicate directly and indirectly via the home agent with the mobile node and 2, proves that mobile node has ownership of both identities as half the tokens are sent via an alternative network to the second identity which combined with the first identity means that the mobile node is the only node capable of receiving all the tokens allowing it to create the binding key. This provides an authentication solution, which is low cost and has minimum resource requirements.

Dual Identity Return Routability has been designed to be incorporated with the distributed authentication protocol however could be used as a stand-alone security solution. This may be useful for distributed mesh networks, which could be formed with the use of WiMAX. However, no matter which transmission technology is used for 4G networks, Dual Identity Return Routability will be compatible because it works on the IP level making it particularly useful for hybrid networks.

5.2.3 Mobile Home Agents

Mobile IPv6 provides two methods of communication between the mobile and correspondent node. The first is triangle routing, which is when all communication to the mobile node is via the home agent. This is necessary as the home agents' IP address is static and is the first point of contact for any communication to the mobile node. The disadvantage however is that the further the mobile node travels from the home agent the further data packets will have to travel to reach their destination.

The second method involves the use of a route optimization technique, which allows direct communication between the mobile and correspondent node. This is achieved with the use of binding updates. The disadvantage to this method is that the location of the mobile node is revealed to any correspondent in communication with it, which could be a potential security risk.
This section introduces an alternative method, which provides the best of both worlds without the disadvantages.

5.2.3.1 Mobile Agents Technology Introduced in to Mobile IPv6

The concept involves the introduction of mobile agent technology into mobile IPv6 networks.
The way they would be used is as an intermediary between the mobile node and the correspondent effectively becoming triangle routing. However the mobile agent would reside on the IPv6 node which the mobile node is using as its point of attachment. The mobile agent is a piece of software responsible for routing messages from other nodes to the mobile node and at the same time provide location privacy by acting as a proxy and masking the true IP address of the mobile node.

As the mobile agent resides on the mobile nodes point of attachment there is negligible latency in comparison to triangle routing via the home agent. As the mobile agent will effectively resume most of the roles of the home agent we can call it a mobile home agent. But why is it mobile?

As it resides on the mobile nodes point of attachment, if the mobile node travels to a new location it will connect to a new point of attachment which will then be responsible for the mobile node as all communications are handed over to it. However the mobile home agent would not lose communication with the mobile node as the software is autonomous and capable of duplicating itself to the new point of attachment and resuming its role in the network. To the Correspondent, the Mobile Home Agent will appear as if it was the Mobile Node, as it changes its IP address with every new location and sends a binding update to the Correspondent.

Every time the mobile node moves to a new point of attachment the mobile home agent will follow providing constant location privacy with the advantages of low latency communication. This process can be seen in Figure 17.

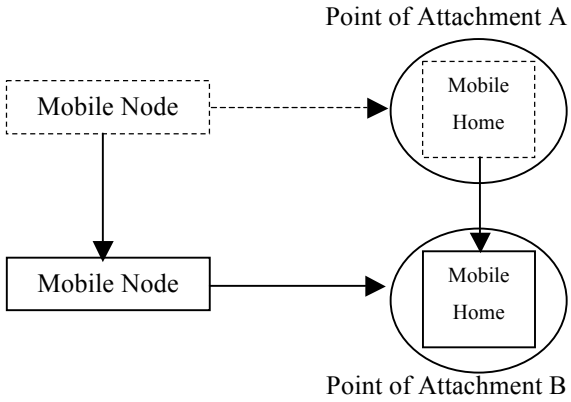

Figure 17. Mobile node and mobile home agent migrating to a new point of attachment.

5.2.3.2 Mobile Home Agent used in a Mobile-to-Mobile Communication.

In this scenario we will assume that the correspondent node is mobile and so requires a mobile home agent.

Message 1.

The mobile node MN attempts to contact the correspondent node CN. The correspondent node however is not directly contactable because it is mobile and it too has a Mobile Home Agent. The Mobile node will have to contact the correspondent's traditional home agent first which will then forward the messages to the correspondent node. The mobile node's care of address CoA and home address HoA are signed using it's private key MNK- and is sent with the Mobile Node's public key, MNK+, to the HA2 Home Agent of the CN the correspondent node. However the CoA care of address given is not the mobile nodes true address, it is the address of its Mobile Home Agent. This is to protect the location of the mobile node. Therefore the proxy care of address which is the Mobile Home Agent is represented by MHA.

In message 3 the correspondent will compare the mobile nodes' public key with the supplied care of address. Under the circumstances this test will fail as the mobile home agents care of address will not match the public key of the mobile node. Therefore the mobile node must supply a public key based on the address of the mobile home agent, MHAK+, and sign the messages with it's private key MHAK-.
All the messages exchanged can be seen in Figure 18.

MN ⟶ HA2: MHAK+, MHAK-(MHA, HoA).

Message 2.

The Correspondent Node's Home Agent received the message from the Mobile Node and forwards it to the correspondent node.

HA2 ⟶ CN: MHAK+, MHAK-(MHA, HoA).

Message 3.

The corresponding node uses the public key MHAK+ to decrypt the MHA and HoA addresses which proves that the sender is the only one who can encrypt and sign the messages. The correspondent then compares the mobile node's public key MHAK+ with that of its claimed CGA address, MHA, and determines if they match. If they do then return routability and device authentication will proceed, otherwise the connection / binding update request is denied. In this case the public key and CGA address are those of the mobile home agent.

The next step the correspondent will perform the home address check and the care of address check.

The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key Kcn only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

$$\text{Home token} = \text{hash} (K_{cn} \mid \text{source address} \mid N_H \mid 0)$$

This is then sent to the home agent.

$$CN \longrightarrow HA: \ HoT.$$

Message 4.

The Home Test packet is then forwarded to the mobile node's care of address. This is sent directly to the mobile node as it is assumed that the home agent is a trusted node and needs to know the location of the mobile node anyway. So sending data via the mobile home agent would have no benefit.

$$HA \longrightarrow MN: \ HoT.$$

Message 5.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 3, however the token generated is slightly different.

$$\text{Care-of token} = \text{hash}(K_{cn} \mid \text{source address} \mid N_C \mid 1)$$

This is then sent directly to the mobile node within a Care of test (CoT) packet. Or so the correspondent thinks. In actuality the correspondent node sends the Care of test (CoT) packet to the mobile home agent.

$$CN \longrightarrow MHA: \ CoT.$$

Message 6.

The mobile home agent tunnels the care of test to the mobile node.

$$MHA \longrightarrow MN: \ CoT.$$

Message 7.

The mobile node receives both tokens from both the test packets sent. It then creates a binding key Kbm by hashing the two tokens together.

$$K_{bm} = \text{hash} (\text{home token} \mid \text{care-of token})$$

The key is used to protect the first and following binding updates. The mobile node then sends a binding update request to the correspondent node, via the correspondents home agent, which is protected with the binding key Kbm.

$$MN \longrightarrow HA2: \ K_{bm}(BU)$$

Message 8.

The Correspondents home agent forwards the binding update request to the correspondent node.

$$HA2 \longrightarrow CN: \ K_{bm}(BU)$$

Message 9.

This is where traditionally the correspondent would decrypt the data and accept the binding update, however before this begins it must wait for the result of another authentication protocol to complete. This authentication takes place simultaneously with return routability.

The correspondent node sends a request message to the mobile node for its authentication data (RAD).

$$CN \longrightarrow MHA: \ RAD$$

Message 10.

The mobile home agent tunnels the request for authentication data (RAD) to the mobile node.

$$MHA \longrightarrow MN: \ RAD$$

Message 11.

The mobile node replies to the message by sending its authentication data, which includes the mobile home agents' current address, its sim number, IMEI number, phone number and even and options for user authentication such as biometric data and time stamps. This is sent to the CN, via HA2, encrypted with the binding key Kbm.

$$MN \longrightarrow HA2: \ K_{bm}$$

(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 12.

The correspondent's mobile home agent, HA2, forwards the encrypted authentication data to the correspondent.

$$HA2 \longrightarrow CN: \ K_{bm}$$

(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 13.

Simultaneously to message 9, the correspondent sends a request for authentication data message to the home agent.

$$CN \longrightarrow HA: \ RAD$$

Message 14.

The home agent does not have the binding key so sending the authentication data would be a security risk. Instead the home agent hashes the authentication data together and sends that to the correspondent via it's home agent.

HA ⟶ HA2: Hash
(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 15.

The correspondents home agent forwards the authentication data to the home agent.

HA2 ⟶ CN: Hash
(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 16.

Now the correspondent will have both the hash of the authentication data and the authentication data encrypted with the binding key. The correspondent performs the authentication comparison by decrypting the binding key and hashing the authentication data received from the mobile node then comparing this to the hash received by the home agent.

If the result of the authentication is successful then the binding update is accepted and a binding acknowledgement BA is sent to the mobile node allowing it to communicate directly with the correspondent via it's mobile home agent, which speeds up the communication and still maintains location privacy.

CN ⟶ MHA: BA

Message 17.

The mobile home agent passes the binding acknowledgement to the mobile node to let it know that the process has been successful.

MHA ⟶ MN: BA

The authentication mechanism is optional and is part of the distributed authentication protocol. The use of mobile home agents can be used on their own, with authentication as seen here or it can be used with the full implementation of the distributed authentication protocol which utilizes dual identity return routability [9]. The above message exchange allows for mobile correspondent nodes to also have their location privacy by implementing their own mobile correspondent home agents which communicate

directly with the mobile nodes mobile home agent, acting as a secure proxy with negligible communication latency.

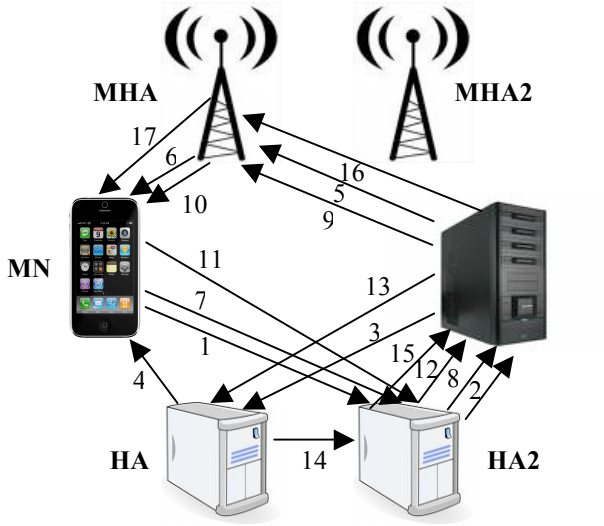

Figure 18. Mobile Home Agent message exchange in mobile-to-mobile communication.

Once the protocol has completed authentication of the mobile and correspondent nodes and the binding update has been exchanged, then direct route optimized communication can take place between the communicating nodes via the Mobile Home Agents on the points of attachment shown in Figure 19. This provides low latency communication with the benefit of a non processor intensive location privacy security solution due to the mobile agent software not running on the mobile device itself but running on the points of attachment.

Figure 19. Communication between mobile-to-mobile nodes via mobile home agents on points of attachment

5.2.3.3 Mobile Home Agent used in a Mobile-to-Static Communication.

The introduction of mobile home agents will noticeably increase the speed of node communication and protect the identity of the mobile nodes' current location.

Firstly all nodes should be using cryptographically generated address, which have been previously created by the function discussed in [2]:

$$(2),\ \text{Host ID} = \text{HASH}_{62}(\text{public key})$$

In this scenario we will assume that the correspondent node is static and so does not require a mobile home agent.

Message 1.

The mobile node MN attempts to contact the correspondent node CN. The mobile node's care of address CoA and home address HoA are signed using it's private key MNK- and is sent with the Mobile Node's public key, MNK+, to CN the correspondent node. Message flows are shown in Figure 20. However the CoA care of address given is not the mobile nodes true address, it is the address of its Mobile Home Agent. This is to protect the location of the mobile node. Therefore the proxy care of address which is the Mobile Home Agent is represented by MHA.

In message 2 the correspondent will compare the mobile nodes' public key with the supplied care of address. Under the circumstances this test will fail as the mobile home agents care of address will not match the public key of the mobile node. Therefore the mobile node must supply a public key based on the address of the mobile home agent, MHAK+, and sign the messages with it's private key MHAK-.

$$\text{MN} \longrightarrow \text{CN:}\ \ \text{MHAK+, MHAK-(MHA, HoA).}$$

Message 2.

The corresponding node uses the public key MHAK+ to decrypt the MHA and HoA addresses which proves that the sender is the only one who can encrypt and sign the messages. The correspondent then compares the mobile node's public key with that of its claimed CGA address and determines if they match. If they do then return routability and device authentication will proceed, otherwise the connection / binding update request is denied. In this case the public key and CGA address are those of the mobile home agent.

The next step the correspondent will perform the home address check and the care of address check.

The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key K_{cn} only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

Home token = hash (K_{cn} | source address | N_H | 0)

This is then sent to the home agent.

CN ⟶ HA: HoT.

Message 3.

The Home Test packet is then forwarded to the mobile node's care of address. This is sent directly to the mobile node as it is assumed that the home agent is a trusted node and needs to know the location of the mobile node anyway. So sending data via the Mobile home agent would have no benefit.

HA ⟶ MN: HoT.

Message 4.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 3, however the token generated is slightly different.

Care-of token = hash(K_{cn} |source address| N_C |1)

This is then sent directly to the mobile node within a Care of test (CoT) packet. Or so the correspondent thinks. In actuality the correspondent node sends the Care of test (CoT) packet to the mobile home agent.

CN ⟶ MHA: CoT.

Message 5.

The mobile home agent tunnels the care of test to the mobile node.

MHA ⟶ MN: CoT.

Message 6.

The mobile node receives both tokens from both the test packets sent. It then creates a binding key K_{bm} by hashing the two tokens together.

K_{bm} = hash (home token | care-of token)

The key is used to protect the first and following binding updates. The mobile node then sends a binding update request to the correspondent node, which is protected with the binding key $K_{bm.}$

MN ⟶ CN: K_{bm}(BU)

Message 7.

This is where traditionally the correspondent would decrypt the data and accept the binding update, however before this begins it must wait for the result of another authentication protocol to complete. This authentication takes place simultaneously with return routability.

The correspondent node sends a request message to the mobile node for its authentication data (RAD).

CN ⟶ MHA: RAD

Message 8.

The mobile home agent tunnels the request for authentication data (RAD) to the mobile node.

MHA ⟶ MN: RAD

Message 9.

The mobile node replies to the message by sending its authentication data, which includes the mobile home agents' current address, its sim number, IMEI number, phone number and even and options for user authentication such as biometric data and time stamps. This sent to the CN encrypted with the binding key $K_{bm.}$

MN ⟶ CN: K_{bm}(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 10.

Simultaneously to message 7, the correspondent sends a request for authentication data message to the home agent.

CN ⟶ HA: RAD

Message 11.

The home agent does not have the binding key so sending the authentication data would be a security risk. Instead the home agent hashes the authentication data together and sends that to the correspondent.

HA ⟶ CN: Hash(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 12.

Now the correspondent will have both the hash of the authentication data and the authentication data encrypted with the binding key. The correspondent performs the authentication comparison by decrypting the binding key and hashing the authentication data received from the mobile node then comparing this to the hash received by the home agent.

If the result of the authentication is successful then the binding update is accepted and a binding acknowledgement BA is sent to the mobile node allowing it to communicate directly with the correspondent.

CN ⟶ MHA: BA

Message 13.

The mobile home agent passes the binding acknowledgement to the mobile node to let it know that the process has been successful.

MHA ⟶ MN: BA

The authentication mechanism is optional and is part of the distributed authentication protocol. The use of mobile home agents can be used on their own, with authentication as seen here or it can be used with the full implementation of the distributed authentication protocol which utilizes dual identity return routability [6] and has support for mobile correspondent nodes which can also have their location privacy by implementing their own mobile correspondent home agent.

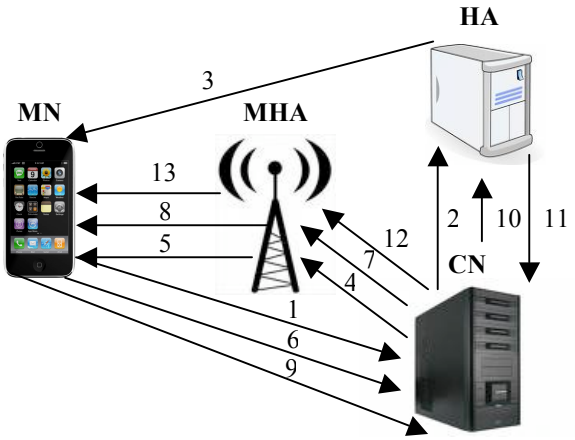

Figure 20. Mobile Home Agent message exchange in mobile-to-static communication.

All the messages exchanged within the mobile home agent mobile-to-static communication can be seen in Figure 20.

5.2.3.4 Summary

This thesis has shown that the Mobile IPv6 route optimization protocol is vulnerable to a variety of attacks which attempt to disrupt or hijack communication between the mobile and the correspondent nodes.

Several security solutions were investigated which were specifically designed to protect location privacy. But the main drawback of these solutions was an increase in latency between communication of the mobile node and the correspondent.

A second technology, mobile agents, were investigated which could potentially change the way networks operate. These are autonomous software based programs which can migrate to another node on the network independently of any other process. They work well in heterogeneous networks and are capable of managing network messages.

This technology was the basis for the proposed security protocol using mobile home agents. Mobile home agents act as a proxy home agent which follows the mobile node as it moves from point of attachment to point of attachment. The mobile home agent resides on the point of attachment itself therefore even though technically the solution reintroduces triangle routing in some respect, in reality there is a negligible latency increase as the data packet would have to pass via the point of attachment anyway to reach the mobile node.

The mobile home agent preserves the mobile nodes location privacy by acting as a proxy and passing all messages to the mobile node via a secure tunnel.
When the mobile node migrates to a new point of attachment the mobile home agent duplicates itself and is transmitted to the new point of attachment when it continues to act as the proxy for the mobile node. The home agent keeps track of both of these entities to ensure they are reachable.
The advantage of the proposed solution is that it is entirely software based and no new hardware would be needed to be introduced, making it a very cost effective option. The location of the mobile node is protected without the cost of increased latency.

The only disadvantages rest with the fact that the mobile home agent is autonomous and so its behavior relies heavily on its robust programming and that every point of attachment may have to be modified to accept mobile agents.
The proposed solution will be tested with the network simulation software Opnet / Omnet. The results will be gathered and compared to other security solutions in terms of effectiveness and impact on latency and resources. It is believed that this proposal will provide a robust and unique security solution.

5.3 Combined Distributed Authentication Security Solution

The security solutions presented can be used individually or can be combined together to form a super security solution. This section describes how combining the distributed authentication protocol, dual identity return routability and mobile home agents together forms the super security solution and demonstrates how it works and how each process interacts with one another.

5.3.1 Combined Solution used in a Mobile-to-Mobile Communication.

The introduction of mobile home agents will noticeably increase the speed of node communication and protect the identity of the mobile nodes' current location.
Firstly all nodes should be using cryptographically generated address, which have been previously created by the function discussed in [2]:

$$\text{Host ID} = \text{HASH}_{62}(\text{public key})$$

In this scenario we will assume that the correspondent node is mobile and so requires a mobile home agent.

Message 1.
The mobile node MN attempts to contact the correspondent node CN. The correspondent node however is not directly contactable because it is mobile and it too has a Mobile Home Agent. The Mobile node will have to contact the correspondent's traditional home agent first which will then forward the messages to the correspondent node. The mobile node's care of address CoA and home address HoA are signed using it's private key MNK- and is sent with the Mobile Node's public key, MNK+, to the HA2 Home Agent of the CN the correspondent node. Message flows are shown in Figure 21. However the CoA care of address given is not the mobile nodes true address, it is the address of its Mobile Home Agent. This is to protect the location of the mobile node. Therefore the proxy care of address which is the Mobile Home Agent is represented by MHA.

In message 3 the correspondent will compare the mobile nodes' public key with the supplied care of address. Under the circumstances this test will fail as the mobile home agents care of address will not match the public key of the mobile node. Therefore the mobile node must supply a public key based on the address of the mobile home agent, MHAK+, and sign the messages with it's private key MHAK-.
All the messages exchanged can be seen in Figure 21.

MN ⟶ HA2: MHAK+, MHAK-(MHA, HoA).

Message 2.

The Correspondent Node's Home Agent received the message from the Mobile Node and forwards it to the correspondent node.

HA2 ⟶ CN: MHAK+, MHAK-(MHA, HoA).

Message 3.

The corresponding node uses the public key MHAK+ to decrypt the MHA and HoA addresses which proves that the sender is the only one who can encrypt and sign the messages. The correspondent then compares the mobile node's public key, which belongs to the Mobile Home Agent, with that of its claimed CGA address and determines if they match. If they do then the new dual identity return routability and device authentication will proceed, otherwise the connection / binding update request is denied. The next step the correspondent will perform the home address check and the care of address check. The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key K_{cn} only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

Home token1 = hash (K_{cn} | source address | N_H | 0)

This is then sent to the home agent.

CN ⟶ HA: HoT1.

Message 4.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT1.

Message 5.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 1 and 2, however the token generated is slightly different

Care-of token1 = hash (K_{cn} | source address | N_C | 1)

This is then sent to the mobile node within a Care of test (CoT) packet, via the Mobile Home Agent, as the Correspondent node believes that is the care of Address.

CN ⟶ MHA ⟶ MN: CoT1.

Message 6.

Home token2 = hash (K_{cn} | source address | N_{H2} | 2)

This is then sent to the home agent of the second identity.

CN ⟶ HA: HoT2.

Message 7.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT2.

Message 8.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 4 and 5, again the token generated is slightly different and the nonce's are also different with N_{H2} and N_{C2} for the second token pair.

Care-of token2 = hash (K_{cn} | source address | N_{C2} | 3)

This is then sent to the mobile node within a Care of test (CoT) packet, via the Mobile Home Agent, as the Correspondent node believes that is the care of Address.

CN ⟶ MHA ⟶ MN: CoT2.

Message 9.

The mobile node receives all four tokens from the four test packets sent. It then creates a binding key K_{bm} by hashing the four tokens together.

K_{bm} = hash (home token | care-of token | home token2 | care-of token2)

The key is used to protect the first and following binding updates. The mobile node then sends a binding update request to the correspondent node, which is protected with the binding key K_{bm}

MN ⟶ HA2: K_{bm}(BU)

Message 10.

The Correspondents home agent forwards the binding update request to the correspondent node.

$$HA2 \longrightarrow CN: \ K_{bm}(BU)$$

Message 11.

This is where traditionally the correspondent would decrypt the data and accept the binding update, however before this begins it must wait for the result of another authentication protocol to complete. This authentication takes place simultaneously with return routability.

The correspondent node sends a request message to the mobile node for its authentication data (RAD).

$$CN \longrightarrow MHA: \ RAD$$

Message 12.

The mobile home agent tunnels the request for authentication data (RAD) to the mobile node.

$$MHA \longrightarrow MN: \ RAD$$

Message 13.

The mobile node replies to the message by sending its authentication data, which includes the mobile home agents' current address, its sim number, IMEI number, phone number and even and options for user authentication such as biometric data and time stamps. This is sent to the CN, via HA2, encrypted with the binding key Kbm.

$$MN \longrightarrow HA2: \ K_{bm}$$

(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 14.

The correspondent's mobile home agent, HA2, forwards the encrypted authentication data to the correspondent.

$$HA2 \longrightarrow CN: \ K_{bm}$$

(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 15.

Simultaneously to message 11, the correspondent sends a request for authentication data message to the home agent.

$$CN \longrightarrow HA: \ RAD$$

Message 16.

The home agent does not have the binding key so sending the authentication data would be a security risk. Instead the home agent hashes the authentication data together and sends that to the correspondent via it's home agent.

HA ⟶ HA2: Hash
(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 17.

The correspondents home agent forwards the authentication data to the home agent.

HA2 ⟶ CN: Hash
(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 18.

Now the correspondent will have both the hash of the authentication data and the authentication data encrypted with the binding key. The correspondent performs the authentication comparison by decrypting the binding key and hashing the authentication data received from the mobile node then comparing this to the hash received by the home agent.

If the result of the authentication is successful then the binding update is accepted and a binding acknowledgement BA is sent to the mobile node allowing it to communicate directly with the correspondent via it's mobile home agent, which speeds up the communication and still maintains location privacy.

CN ⟶ MHA: BA

Message 19.

The mobile home agent passes the binding acknowledgement to the mobile node to let it know that the process has been successful.

MHA ⟶ MN: BA

The authentication mechanism is optional and is part of the distributed authentication protocol. The use of mobile home agents can be used on their own, with authentication as seen here or it can be used with the full implementation of the distributed authentication protocol which utilizes dual identity return routability [9]. The above message exchange allows for mobile correspondent nodes to also have their location privacy by implementing their own mobile correspondent home agents which communicate directly with the mobile nodes mobile home agent, acting as a secure proxy with negligible communication latency.

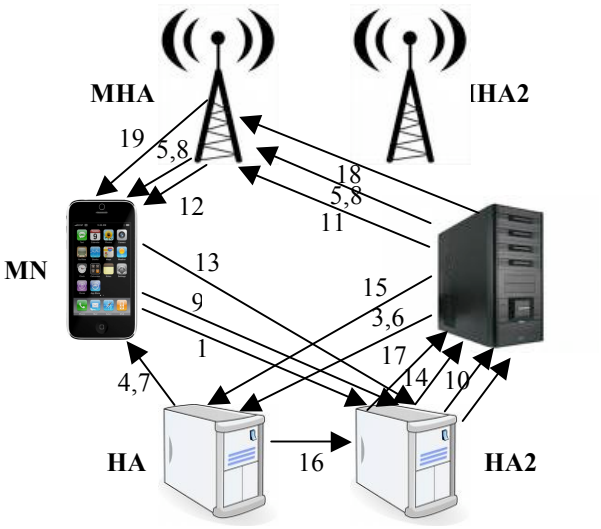

Figure 21. Combined Solution message exchange in mobile-to-mobile communication.

Once the protocol has completed authentication of the mobile and correspondent nodes and the binding update has been exchanged, then direct route optimized communication can take place between the communicating nodes via the Mobile Home Agents on the points of attachment shown in Figure 21. This provides low latency communication with the benefit of a non processor intensive location privacy security solution due to the mobile agent software not running on the mobile device itself but running on the points of attachment.

5.3.2 Combined Solution used in a Mobile-to-Static Communication.

The introduction of mobile home agents will noticeably increase the speed of node communication and protect the identity of the mobile nodes' current location.
Firstly all nodes should be using cryptographically generated address, which have been previously created by the function discussed in [2]:

$$\text{Host ID} = \text{HASH}_{62}(\text{public key})$$

In this scenario we will assume that the correspondent node is static and so does not require a mobile home agent.

Message 1.
The mobile node MN attempts to contact the correspondent node CN. The mobile node's care of address CoA and home address HoA are signed using it's private key MNK- and is sent with the Mobile Node's public key, MNK+, to CN the correspondent node. Message flows are shown in Figure 22. However the CoA care of address given is not the mobile nodes true address, it is the address of its

Mobile Home Agent. This is to protect the location of the mobile node. Therefore the proxy care of address which is the Mobile Home Agent is represented by MHA.
In message 2 the correspondent will compare the mobile nodes' public key with the supplied care of address. Under the circumstances this test will fail as the mobile home agents care of address will not match the public key of the mobile node. Therefore the mobile node must supply a public key based on the address of the mobile home agent, MHAK+, and sign the messages with it's private key MHAK-.

MN ⟶ CN: MHAK+, MHAK-(MHA, HoA).

The Correspondent node (CN) will test to see if the Mobile node MN is reachable at the care of address and also test the other identity address is reachable. The Mobile node has two identities which are linked together making spoofing a lot more difficult and proving the user is the owner of the identity.

Message 2.

The corresponding node uses the public key MHAK+ to decrypt the MHA and HoA addresses which proves that the sender is the only one who can encrypt and sign the messages. The correspondent then compares the mobile node's public key, which belongs to the Mobile Home Agent, with that of its claimed CGA address and determines if they match. If they do then the new dual identity return routability and device authentication will proceed, otherwise the connection / binding update request is denied.
The correspondent will perform the home address check and the care of address check. The correspondent will send a home test (HoT) packet, which is assumed that the home agent will tunnel to the mobile node. The HoT packet consists of a home keygen token generated by hashing the secret key K_{cn} only known to the correspondent. A nonce index N_H is also included to allow the CN to find the appropriate nonce easily.

Home token1 = hash (K_{cn} | source address | N_H | 0)

This is then sent to the home agent.
CN ⟶ HA: HoT1.

Message 3.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶ MN: HoT1.

Message 4.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 2 and 3, however the token generated is slightly different

Care-of token1 = hash (K_{cn} | source address | N_C | 1)

This is then sent directly to the mobile node within a Care of test (CoT) packet, via the Mobile Home Agent.

CN ⟶MHA⟶MN: CoT1.

Message 5.

Home token2 = hash (K_{cn} | source address | N_{H2} | 2)

This is then sent to the home agent of the second identity.

CN ⟶HA: HoT2.

Message 6.

The Home Test packet is then forwarded to the mobile node's care of address.

HA ⟶MN: HoT2.

Message 7.

The correspondent also performs a care of address test (CoT), which is similar to the home address test and takes place at the same time as message 5 and 6, again the token generated is slightly different and the nonce's are also different with N_{H2} and N_{C2} for the second token pair.

Care-of token2 = hash (K_{cn} | source address | N_{C2} | 3)

This is then sent directly to the mobile node within a Care of test (CoT) packet, via the Mobile Home Agent.

CN ⟶MHA⟶ MN: CoT2.

Message 8.

The mobile node receives all four tokens from the four test packets sent. It then creates a binding key K_{bm} by hashing the four tokens together.

K_{bm} = hash (home token | care-of token | home token2 | care-of token2)

The key is used to protect the first and following binding updates. The mobile node then sends a binding update request to the correspondent node, which is protected with the binding key K_{bm}

MN ⟶ CN: K_{bm}(BU)

This protocol proves that the mobile node is reachable at its current address, preventing denial of service attacks and proves that the two identities are associated with each other proving ownership of the phone numbers / IP addresses and providing a cheap method of authentication.

Message 9.

This is where traditionally the correspondent would decrypt the data and accept the binding update, however before this begins it must wait for the result of another authentication protocol to complete. This authentication takes place simultaneously with the Dual Identity Return Routability.
The correspondent node sends a request message to the mobile node for its authentication data (RAD).

CN ⟶ MHA: RAD

Message 10.

The mobile home agent tunnels the request for authentication data (RAD) to the mobile node.

MHA ⟶ MN: RAD

Message 11.

The mobile node replies to the message by sending its authentication data, which includes the mobile home agents' current address, its sim number, IMEI number, phone number and even and option for user authentication such as biometric data. This sent to the CN encrypted with the binding key $K_{bm.}$

MN ⟶ CN: K_{bm}(MHA, Sim No, IMEI, Phone No., Biometric)

Message 12.

Simultaneously to message 9, the correspondent sends a request for authentication data message to the home agent.

CN ⟶ HA: RAD

Message 13.

The home agent does not have the binding key so sending the authentication data would be a security risk. Instead the home agent hashes the authentication data together and sends that to the correspondent.

HA ⟶ CN: Hash(MHA, Sim No, IMEI, Phone No., Biometric, Timestamp)

Message 14.

Now the correspondent will have both the hash of the authentication data and the authentication data encrypted with the binding key. The correspondent performs the authentication comparison by decrypting the binding key and hashing the authentication data received from the mobile node then comparing this to the hash received by the home agent.

If the result of the authentication is successful then the binding update is accepted and a binding acknowledgement BA is sent to the mobile node allowing it to communicate directly with the correspondent.

CN ⟶ MHA: BA

Message 15.

The mobile home agent passes the binding acknowledgement to the mobile node to let it know that the process has been successful.

MHA ⟶ MN: BA

The Super Solution combines the three unique contributions, which have been presented in this thesis. It combines the Distributed Authentication protocol, which allows for user authentication and allows for the processing of the data to be done on a more powerful external node. The Super Solution also uses Dual Identity Return Routability which allows mobile nodes to have two separate but linked identities which allows for protection against identity spoofing. And finally the Super Solution utilizes Mobile Home Agents, which is a software mobile agent proxy for the mobile node providing location privacy without increase in communication latency. This solution can be used in a mobile to static node configuration as demonstrated above and has support for mobile correspondent nodes, which can also have their location privacy by implementing their own mobile correspondent home agent.

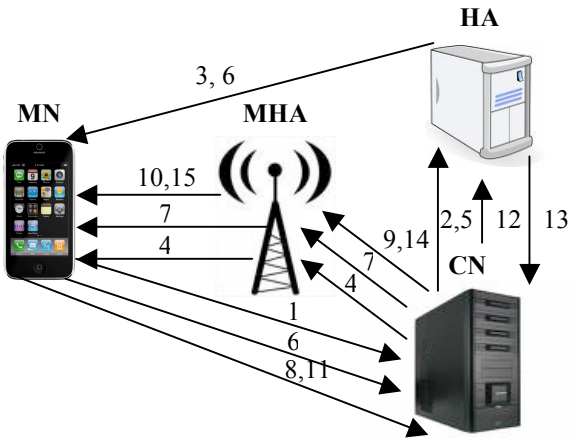

Figure 22. Combined Solution message exchange in mobile-to-static communication.

All the messages exchanged within the mobile home agent mobile to static communication can be seen in Figure 22.

5.4 Proposed Solution Conclusion

The proposed solution is based on three unique security implementations. Cryptographically Generated Addresses, Return Routability and Authentication Verification.
The first two technologies are well-established techniques. Cryptographically Generated Addresses provide a reasonable assurance that the address of the user is indeed owned by them and not spoofed. Return Routability provides location authentication proving the communicating device is at the IP address claimed and again combats spoofing.
The third aspect of the security protocol provides solid device authentication and can be expanded to include user authentication in case of device theft.

Adding security features means that there will be an increase in processing power needed by devices. To resolve this issue the protocol proposes using a distributed authentication architecture. The home agent itself will perform part of the processing of the authentication data. This should provide several benefits such as lowering the overall time for the authentication to complete, as different parts of the authentication would occur on the mobile node and at the home agent.

The advantages of using a distributed authentication protocol is that there is a predicted increase in processing speed concerning the completion of security techniques which at the same time not over burdening the mobile processor with all the work.
The disadvantage is that there is an increase in network traffic, however optimisation to the protocol may be able to reduce this.

The research attempted to discover the different types of future technologies in development, which may be incorporated into the 4G fourth Generation mobile network. One technology stood out which allows multiple phone numbers, or identities, to be simultaneously used on a single sim card. Different wireless transmission technologies were also investigated such as WI-FI and WiMAX. Taking advantage of these technologies, a new security solution was created based on return routability. Secret tokens are sent to the addresses of the mobile node and the home agent of both identities. This provides reasonable reassurance of two things, 1, the mobile node is indeed in the location it claims to be and 2, proves that mobile node has ownership of both identities providing a cheap authentication solution. Dual Identity Return Routability has been designed to be incorporated with the distributed authentication protocol however could be used as a stand-alone security solution. This may be useful for distributed mesh networks, which could be formed with the use of WiMAX. However, no matter which transmission technology is used for 4G networks, Dual Identity Return Routability will be compatible because it works on the IP level making it particularly useful for hybrid networks.

A number of attacks exist which attempt to disrupt or hijack communication by exploiting the vulnerability of the Mobile IPv6 route optimization protocol between the mobile and the correspondent nodes.

The location of the mobile node is sensitive data that can be used to mount attacks. Security solutions, which protect the location data, were researched however these solutions increased the communication latency between the communicating nodes.

Mobile agents technology was used as the basis for the proposed security protocol as it could potentially change the way networks operate. Mobile home agents are autonomous software based programs, which act as a proxy home agent, which follows the mobile node as it moves from point of attachment to point of attachment. Mobile agents can migrate to another node on the network independently of any other process making them suitable to this task.
Even though the solution reintroduces triangle routing, there is a negligible latency increase as the mobile home agent resides on the point of attachment therefore data packets would have to pass via the point of attachment to reach the mobile node.

Mobile agents work well in heterogeneous networks and are capable of managing network messages, this allows the location privacy of the mobile node to remain protected by the mobile home agent as it acts as a proxy, passing all messages to the mobile node via a secure tunnel.

The mobile home agent duplicates itself when migrating from the mobile node to a new point of attachment and is transmitted to the new point of attachment where it continues to act as the proxy for the mobile node. These entities are monitored by the home agent to ensure they are reachable.

The advantage of the proposed solution is that the location of the mobile node is protected without an increase in communication latency, it is entirely software based and no new hardware needs to be introduced, making it a very cost effective option.
The only disadvantages is that every point of attachment may have to be modified to accept mobile agents and the behaviour of the mobile home agent relies heavily on robust A.I programming as they autonomously.

The proposed solution will be tested with the network simulation software Omnet. The results will be gathered and compared to other security solutions in terms of effectiveness and impact on latency and resources. The next chapter will provide the evidence that the proposed unique security solution is robust and effective in protecting the binding updates on the mobile node and this is backed up by the results in section 6.5 of the simulations conducted.

6. Simulation

6.1 Introduction

The proposed solution was programmed with a network simulator to test the effectiveness against attacks from rogue elements. The simulation software chosen was Omnet for it's open source approach and readily available documentation and tutorials. The full C++ source code of the programmed simulations is provided in the appendix.

The simulation was programmed in such a way that it was possible to reconfigure the layout and connections of the nodes without having to reprogram each time. All that was required was to modify the appropriate variables and the relevant code would perform the required tasks.

To test the solutions in a meaningful way it was decided to first take control readings of the network to see how it behaved without security or attacks. This gives us a baseline on which to compare all other results. Next, attacks will take place to see the kind of damage that occurs. The attacks will attempt to impersonate the mobile node and redirect data to the attacker.

The next phase will be to test each individual component of currently available security solutions, namely Cryptographically Generated Addresses and Return Routability, which has been incorporated in to the proposed security solution. These will be tested with a variety of attacks, some specific to the security solution.

We will then test each individual component of the proposed solution with the same attacks, which will allow us to determine effectiveness, network usage and latency.

The next step would be to combine the solutions in various combinations and test cumulative effectiveness against the attacks.

And finally, all these tests will be performed again but this time with the final proposed addition to the security solution, Mobile Home Agents. This will allow us to determine how communication, attacks and security are effected by this significant network modification.

6.2 Why Omnet++ Network Simulation Software?

When considering which simulation software to use to build and model the Mobile IPv6 Network, nodes, attacker and security protocols, the most obvious candidate was either NS2 or Opnet modeller. NS2 was command line only and did not present any form of graphic interface so this was ruled out.

Opnet has a very attractive graphical user interface, however it is a licensed software which is a very controlled item to possess. Initially I had started to learn this software but the licence then ran out. I had it renewed but the second issue I had was that there was very limited documentation on how to use the environment to build what was required.

These combining factors led me to look elsewhere for a simulation software which did not have licensing restrictions, had sufficient documentation available and an attractive graphical interface. All of these requirements were fulfilled with Omnet, an open source modeller and represents a framework approach. It is very popular in academia for its extensibility, due to its open source model, and plentiful online documentation.

The programming language of Omnet is C++ which is useful as I had studied it for my degree.
The simulation was built upon one of the tutorial examples called TicToc15 in which messages are passed back and forth between nodes. This example codes was used as the building blocks for the simulation. The simulation example demonstrates how Omnet can be used to model the behaviour of communicating nodes. The example shows a selection of nodes, one of which sends a packet to the next node, which then selects another node at random and passes the packet on to that node. This was very useful as it demonstrates how the nodes work and so can be modified with the desired behaviours necessary to simulate the proposed solution. The example also provides tools for collecting and displaying data gathered from the nodes. This too proved valuable as the ability to gather data is necessary for analysis of the proposed solution.

The components of an Omnet simulation consist of an NED file, which is the definition of the network. Here on the nodes and all the connections between them are defined. The next component is the .CC file which is the main body of the simulation code which defines the behaviour of the nodes and how they handle and forward the messages. And finally there is the .msg class, which was automatically generated by the tutorial simulation and defines the message packet sent between the nodes.

The heart of the code lies in the .CC file. This is where all the node and network variable are initialised and run. The first message is a self-message to the Mobile Node from it's self. Once a message arrives at a node one of three main functions are called.

The first function is handleMessage(). Here is a list of if / else statements, which test the current state of the network and react accordingly. If a message arrives at a node which does not match its destination address then the forwardMessage() function is called and the packet is routed to the next node in the network. If the packet arrives at the correct destination then depending on the delivered message the appropriate message is created in the generateMessage() function and then forwarded on to the senders address.

As the source code for the simulations tested here are over 60 pages long they will not be placed in the main body of the thesis. However you can find the full code in Appendix C.

6.3 Network Layout

As there are various network configurations each of these will be tested. They consist of:

1. Mobile Node on the Home Network and a static wired Correspondent Node.
2. Mobile Node on a foreign network and a static wired Correspondent Node.
3. Mobile Node on the Home Network and a mobile wireless Correspondent node.
4. Mobile Node on a foreign network and a mobile wireless Correspondent node.

The attacks also will take place in a direct attack configurations to the effected nodes.

There will be a variety of attacks but they are specific to the security solution being employed so they will be discussed in the relevant section.

The results will be gathered and analysed in the forms of data and charts and then conclusions will be drawn.

The network layout:

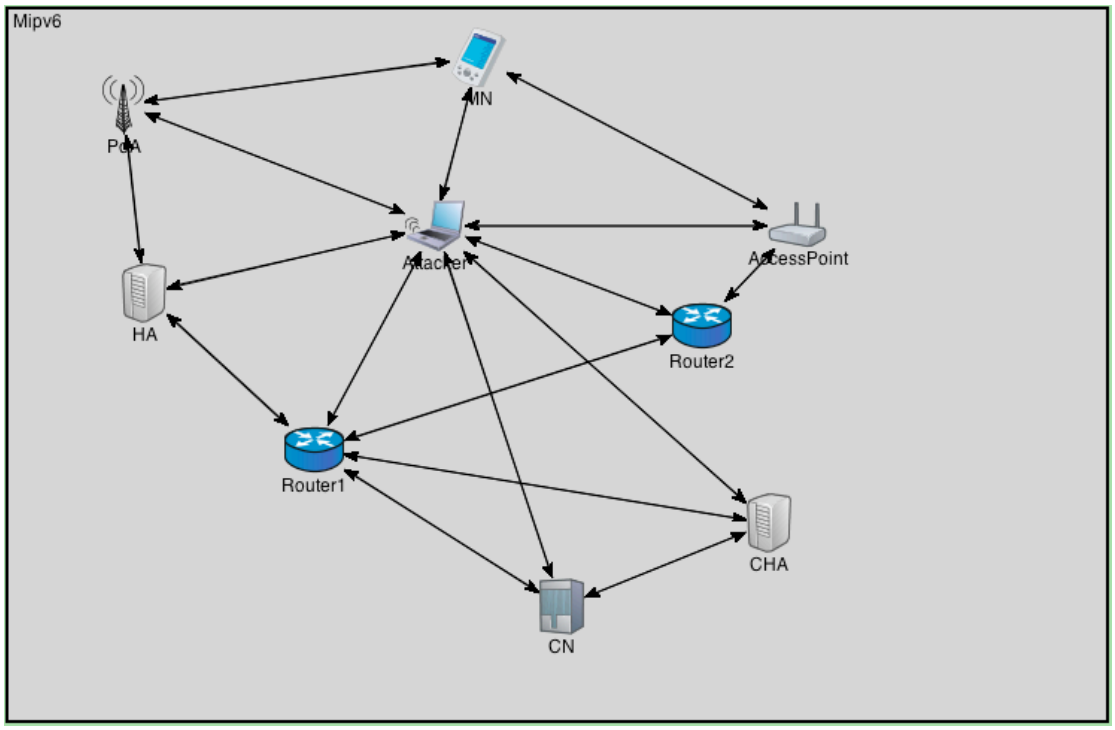

Figure 24. Simulation Network Layout

- MN – Mobile Node
- PoA – Point of Attachment
- HA – Home Agent
- Router1 – Internet Network Router
- CN – Correspondent Node
- CHA – Correspondents Home Agent
- Router2 - Internet Network Router
- Access Point – Wireless Access point on the foreign network that the Mobile node can connect to.

Network configurations:

1. Mobile Node on the Home Network communicating with a static wired Correspondent Node.

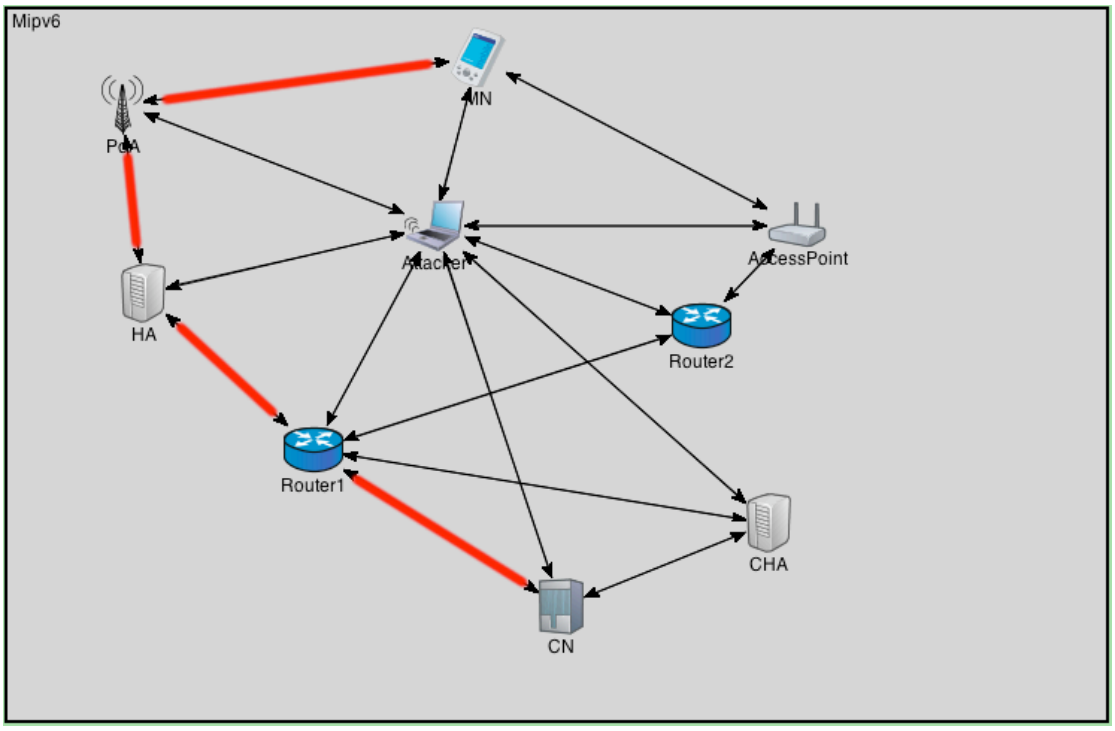

Figure 25. Simulation Network Configuration 1

2. Mobile Node on a foreign network communicating with a static wired Correspondent Node.

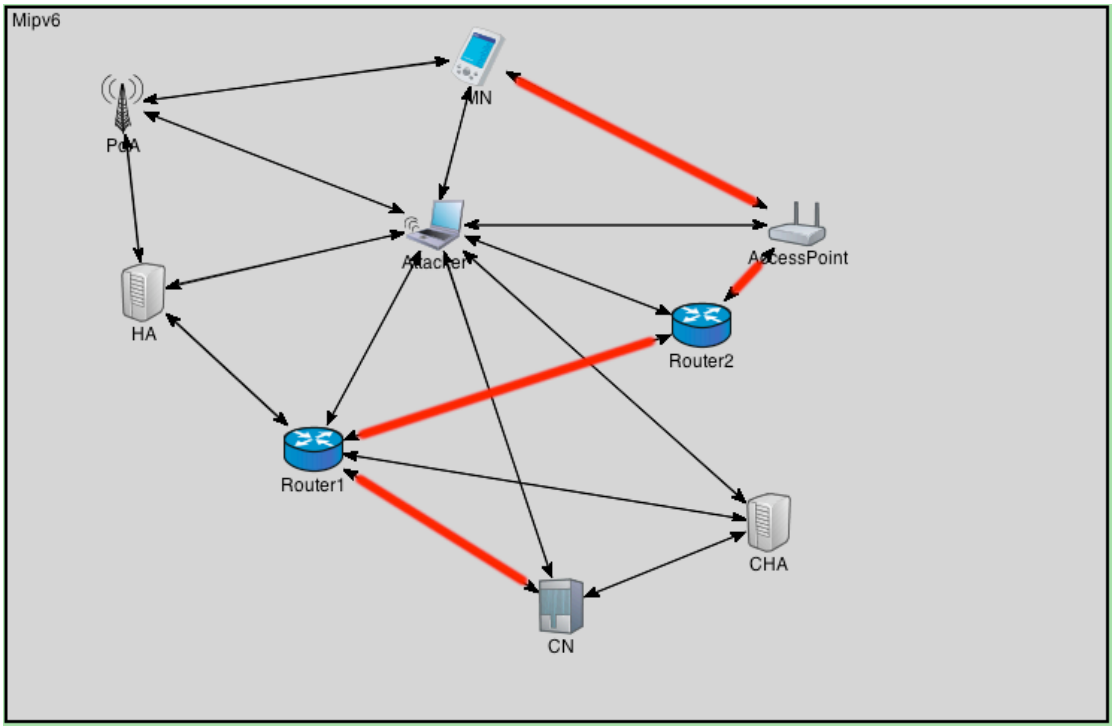

Figure 26. Simulation Network Configuration 2

3. Mobile Node on the Home Network communicating with a mobile wireless Correspondent node.

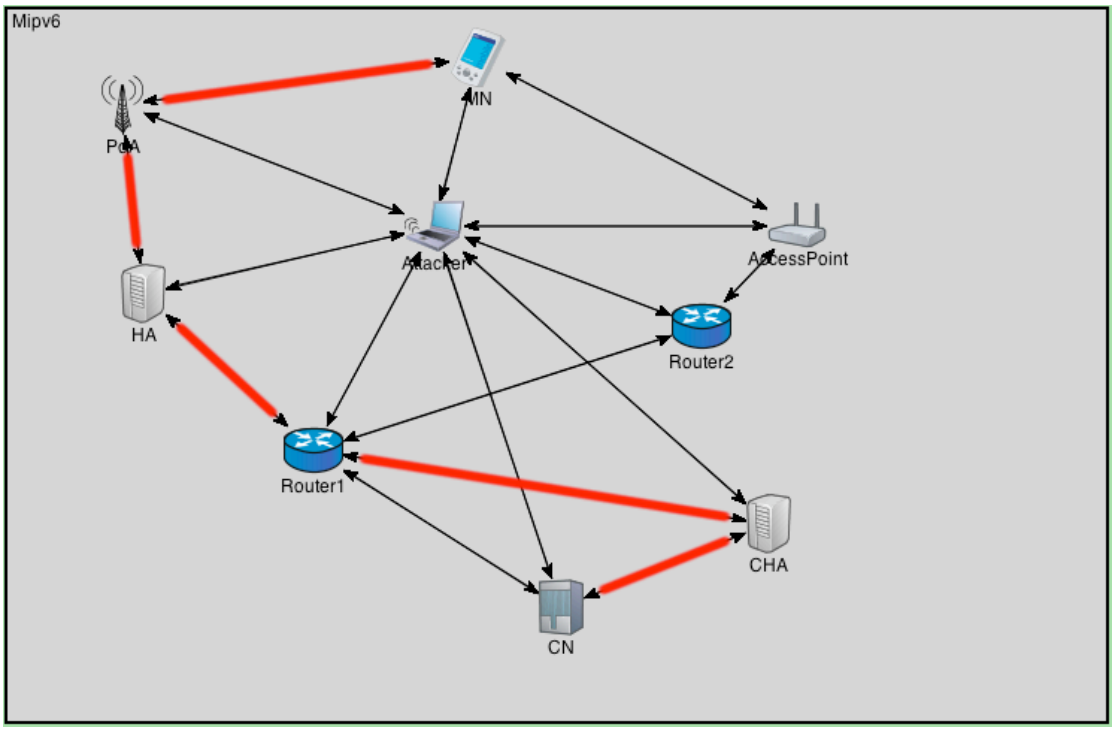

Figure 27. Simulation Network Configuration 3

4. Mobile Node on a foreign network communicating with a mobile wireless Correspondent node.

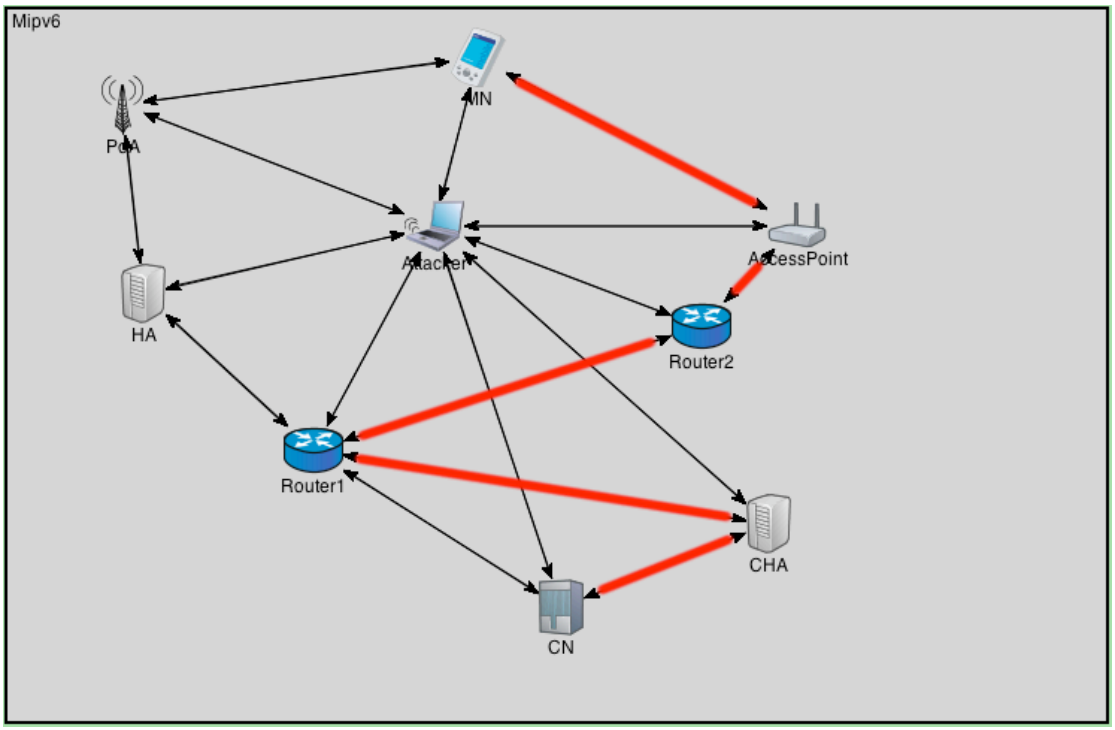

Figure 28. Simulation Network Configuration 4Attack configuration:

Direct attack to the effected nodes.

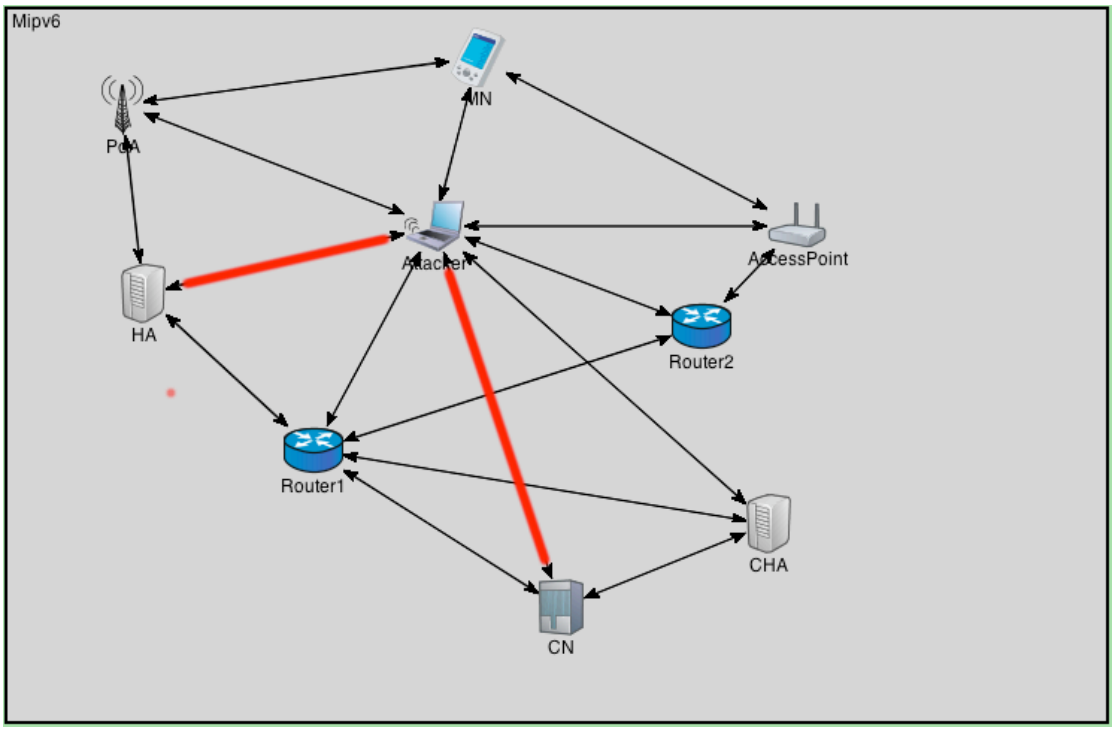

Figure 29. Simulation Attack Configuration

6.4 Simulation Tests

Each Simulation will be run for 60 seconds in simulation time for the results to be comparable.
Also to limit any other external factors from influencing the results, the network connections and nodes all behave with the same level of delay (100ms) and processing speed. This creates a level playing field to calculate latency and effectiveness.

To simulate potential opportunities for attack another binding update will be sent after 5 data packets have been exchanged. If the Attacker or Mobile Node has had no reply within 2 seconds it will resend the connection request.
If the attacker hijacks communication it will delete any duplicate packets to have only a single ongoing dialogue and to reduce network load. The first messages that the nodes send are not included in the results.

The nodes, Attacker and Mobile Node, begin by sending a Request message to the Correspondent Node. This is replied to with an Acknowledgement message and then once that is received the node will send a Binding Update request. For the Mobile to complete the request the Correspondent must reply with a Binding acknowledgement message. However if the Attacker can spoof the Mobile Nodes address and identity it received the binding Acknowledgement it can update the location on the Mobile Node to the Home Agent redirecting all traffic to the Attacker.
To prevent this various existing and proposed security solutions will be implemented to stop the Attacker from impersonating, hijacking, redirecting data and preventing denial of service to the Mobile Node.

Each solution will be tested with the network on its own and with a variety of attacks in differing network configurations.
The main attack to be performed will be a false Binding Update.
However as currently existing security techniques are tested, then so are current methods of bypassing them. Then the proposed security solutions are tested with the same attacks so if they can withstand them.

The following is a list of the simulations that will be run, which will cover the network in its various configurations, different attacks and the security solutions:

Simulation 1: Control. Standard network in 4 configurations for based line establishment.
4 Simulations. Done.
Simulation 2: Control. Standard network attacked in 4 configurations with direct attack methods.
4 Simulations Done
Simulation 3: CGA impact on standard network in 4 configurations.
4 simulations.done
Simulation 4: CGA impact on attacker using its own CGA address.
4 simulations. done
Simulation 5: CGA impact on attacker attempting to spoof the MNs Home Agent.
4 simulations. done
Simulation 6: RR on control network.
4 Simulations done
Simulation 7: RR with attacker using HA as HoT
4 Simulations done
Simulation 8: RR with attacker using it's self to spoof the HA.
4 Simulations. done
Simulation 9: DAP with control network.
4 simulations. done
Simulation 10: DAP with attack
4 simulations.
Simulation 11: DIRR with control network.
4 simulations.
Simulation 12: DIRR with 1st attack.
4 simulations
Simulation 13: DIRR with 2nd attack.
4 simulations
Combined Simulations.
Simulation 14: CGA+RR – Control
4 Simulations done
Simulation 15: CGA+RR – cga 1 rr 1
4 Simulations done
Simulation 16: CGA+RR – cga 2 rr 1
4 Simulations done
Simulation 17: CGA+RR – cga 1 rr 2
4 Simulations done
Simulation 18: CGA+RR – cga 2 rr 2
4 Simulations done
Simulation 19: CGA+RR+DAP – Control
4 Simulations
Simulation 20: CGA+RR+DAP – cga 1 rr 1
4 Simulations done
Simulation 21: CGA+RR+DAP – cga 2 rr 1
4 Simulations done
Simulation 22: CGA+RR+DAP – cga 1 rr 2
4 Simulations done
Simulation 23: CGA+RR+DAP – cga 2 rr 2
4 Simulations done
Simulation 24: CGA+DIRR+DAP – Control
4 Simulations done
Simulation 25: CGA+ DIRR +DAP – cga 1 rr 1
4 Simulations done
Simulation 26: CGA+ DIRR +DAP – cga 2 rr 1
4 Simulations done
Simulation 27: CGA+ DIRR +DAP – cga 1 rr 2
4 Simulations done
Simulation 28: CGA+ DIRR +DAP – cga 2 rr 2
4 Simulations done

Introduction of Mobile Home Agent
(only works with Mobile node on the foreign network so only 2 network configurations are tested):
Simulation 29: MHA – Control
2 Simulations done
Simulation 30: MHA – Attack
2 Simulations done
Simulation 31: CGA impact on standard network in 2 configurations with MHA.
2 simulations. done
Simulation 32: CGA impact on attacker using its own CGA address with MHA.
2 simulations. done
Simulation 33: CGA impact on attacker attempting to spoof the MNs Home Agent with MHA.
2 simulations. Done.
Simulation 34: RR on control network with MHA.
2 Simulations done
Simulation 35: RR with attacker using HA as HoT with MHA
2 Simulations done.
Simulation 36: RR with attacker using it's self to spoof the HA with MHA.
2 Simulations. done
Simulation 37: DAP with control network with MHA.
2 simulations. Done.
Simulation 38: DAP with attack with MHA
2 simulations. done
Simulation 39: DIRR with control network with MHA.
2 simulations. done
Simulation 40: DIRR with 1st attack with MHA.
2 simulations done
Simulation 41: DIRR with 2nd attack with MHA.
2 simulations done
Combined Simulations with Mobile Home Agent.
Simulation 42: CGA+RR with MHA. – Control
2 Simulations done
Simulation 43: CGA+RR with MHA. – cga 1 rr 1
2 Simulations done
Simulation 44: CGA+RR with MHA.– cga 2 rr 1
2 Simulations done
Simulation 45: CGA+RR with MHA. – cga 1 rr 2
2 Simulations done
Simulation 46: CGA+RR with MHA. – cga 2 rr 2
2 Simulations done
Simulation 47: CGA+RR+DAP with MHA. – Control
2 Simulations done
Simulation 48: CGA+RR+DAP with MHA. – cga 1 rr 1
2 Simulations done
Simulation 49: CGA+RR+DAP with MHA. – cga 2 rr 1
2 Simulations done
Simulation 50: CGA+RR+DAP with MHA.– cga 1 rr 2
2 Simulations done
Simulation 51: CGA+RR+DAP with MHA.– cga 2 rr 2
2 Simulations done
Simulation 52: CGA+DIRR+DAP with MHA. – Control
2 Simulations done
Simulation 53: CGA+ DIRR +DAP with MHA. – cga 1 rr 1
2 Simulations done
Simulation 54: CGA+ DIRR +DAP with MHA. – cga 2 rr 1
2 Simulations done
Simulation 55: CGA+ DIRR +DAP with MHA. – cga 1 rr 2
2 Simulations done
Simulation 56: CGA+ DIRR +DAP with MHA. – cga 2 rr 2
2 Simulations done

Table 1. Table of Simulations Run

The scenarios shown in Table 1 are explained here in more detail:

The first simulation of every scenario is a control to gather data that can then be compared to other scenarios.

Controls

Simulation 1: Control. Standard network in 4 configurations for based line establishment.

Simulation 2: Control. Standard network attacked in 4 configurations with direct attack methods.

Cryptographically Generated Addresses

Simulation 3: Cryptographically Generated Addresses (CGA) impact on standard network in 4 configurations.

Simulation 4: CGA impact on attacker using its own CGA address. This Attack is called CGA1.

Simulation 5: CGA impact on attacker attempting to spoof the MNs Home Agent. This Attack is called CGA2.

Return Routability

Simulation 6: Return Routability (RR) on control network.

Simulation 7: RR with attacker using HA as HoT. This Attack is called RR1.

Simulation 8: RR with attacker using it's self to spoof the HA. This Attack is called RR2.

Distributed Authentication Protocol

Simulation 9: Distributed Authentication Protocol (DAP) with control network.

Simulation 10: DAP with attack

Dual Identity Return Routability

Simulation 11: Dual Identity Return Routability (DIRR) with control network.

Simulation 12: DIRR with 1st attack.

Simulation 13: DIRR with 2nd attack.

Combined Simulations.

What you see blow is the simulation number, the security combination used and the attacks mounted:

Simulation 14: CGA+RR – Control

Simulation 15: CGA+RR – cga attack 1 and rr attack 1

Simulation 16: CGA+RR – cga attack 2 rr attack 1

Simulation 17: CGA+RR – cga 1 rr 2

Simulation 18: CGA+RR – cga 2 rr 2

Simulation 19: CGA+RR+DAP – Control

Simulation 20: CGA+RR+DAP – cga 1 rr 1

Simulation 21: CGA+RR+DAP – cga 2 rr 1

Simulation 22: CGA+RR+DAP – cga 1 rr 2

Simulation 23: CGA+RR+DAP – cga 2 rr 2

Simulation 24: CGA+DIRR+DAP – Control

Simulation 25: CGA+ DIRR +DAP – cga 1 rr 1

Simulation 26: CGA+ DIRR +DAP – cga 2 rr 1

Simulation 27: CGA+ DIRR +DAP – cga 1 rr 2

Simulation 28: CGA+ DIRR +DAP – cga 2 rr 2

Introduction of Mobile Home Agent

What you see blow is the simulation number, the security combination used and the attacks mounted:

Simulation 29: Mobile Home Agent (MHA) – Control

Simulation 30: MHA – Attack

Simulation 31: CGA impact on standard network in 2 configurations with MHA.

Simulation 32: CGA impact on attacker using its own CGA address with MHA.

Simulation 33: CGA impact on attacker attempting to spoof the MNs Home Agent with MHA.

Simulation 34: RR on control network with MHA.

Simulation 35: RR with attacker using HA as HoT with MHA

Simulation 36: RR with attacker using it's self to spoof the HA with MHA.

Simulation 37: DAP with control network with MHA.

Simulation 38: DAP with attack with MHA

Simulation 39: DIRR with control network with MHA.

Simulation 40: DIRR with 1st attack with MHA.

Simulation 41: DIRR with 2nd attack with MHA.

Combined Simulations with Mobile Home Agent.

What you see blow is the simulation number, the security combination used with The Mobile Home Agent and the attacks mounted:

Simulation 42: CGA+RR with MHA. – Control

Simulation 43: CGA+RR with MHA. – cga 1 rr 1

Simulation 44: CGA+RR with MHA.– cga 2 rr 1

Simulation 45: CGA+RR with MHA. – cga 1 rr 2

Simulation 46: CGA+RR with MHA. – cga 2 rr 2

Simulation 47: CGA+RR+DAP with MHA. – Control

Simulation 48: CGA+RR+DAP with MHA. – cga 1 rr 1

Simulation 49: CGA+RR+DAP with MHA. – cga 2 rr 1

Simulation 50: CGA+RR+DAP with MHA.– cga 1 rr 2

Simulation 51: CGA+RR+DAP with MHA.– cga 2 rr 2

Simulation 52: CGA+DIRR+DAP with MHA. – Control

Simulation 53: CGA+ DIRR +DAP with MHA. – cga 1 rr 1

Simulation 54: CGA+ DIRR +DAP with MHA. – cga 2 rr 1

Simulation 55: CGA+ DIRR +DAP with MHA. – cga 1 rr 2

Simulation 56: CGA+ DIRR +DAP with MHA. – cga 2 rr 2

6.5 Results

Simulation results Packets Sent/Received in 60 Seconds

Network config:
MNH - CNH= Mobile Node at home to Correspondent Node at home
MNA - CNH= Mobile Node at Away from home to Correspondent Node at home
MNH - CNA= Mobile Node at home to Correspondent Node away
MNA - CNA= Mobile Node at Away from home to Correspondent Node away

Nodes:
MN = Mobile Node
CN = Corresponding Node
ATK = Attacker
HA = Home Agent
MHA = Mobile Home Agent

					Packets Sent					Packets Received						
Sim. No.	Network Config	Attack type	Security Type		MN	CN	ATK	HA	MHA	MN	CN	ATK	HA	MHA	Attacker	Security
1.1	MNH - CNH	None	None	Simulation 1.1	74	75	0	1	0	74	75	0	14	0	none	none
1.2	MNA - CNH	None	None	Simulation 1.2	74	75	0	1	0	74	75	0	14	0		
1.3	MNH - CNA	None	None	Simulation 1.3	59	60	0	1	0	59	60	0	11	0		
1.4	MNA - CNA	None	None	Simulation 1.4	59	60	0	1	0	59	60	0	11	0		
2.1	MNH - CNH	False BU	None	Simulation 2.1	29	331	300	30	0	0	331	330	90	0	pass	none
2.2	MNA - CNH	False BU	None	Simulation 2.2	29	331	300	30	0	0	331	330	90	0		
2.3	MNH - CNA	False BU	None	Simulation 2.3	29	330	299	30	0	0	330	329	90	0		
2.4	MNA - CNA	False BU	None	Simulation 2.4	29	330	299	30	0	0	330	329	90	0		
3.1	MNH - CNH	None	CGA	Simulation 3.1	74	75	0	1	0	74	75	0	14	0	none	working
3.2	MNA - CNH	None	CGA	Simulation 3.2	74	75	0	1	0	74	75	0	14	0		
3.3	MNH - CNA	None	CGA	Simulation 3.3	59	60	0	1	0	59	60	0	11	0		
3.4	MNA - CNA	None	CGA	Simulation 3.4	59	60	0	1	0	59	60	0	11	0		
4.1	MNH - CNH	Own CGA	CGA	Simulation 4.1	29	331	300	30	0	0	331	330	90	0	pass	bypased
4.2	MNA - CNH	Own CGA	CGA	Simulation 4.2	29	331	300	30	0	0	331	330	90	0		
4.3	MNH - CNA	Own CGA	CGA	Simulation 4.3	29	330	299	30	0	0	330	329	90	0		
4.4	MNA - CNA	Own CGA	CGA	Simulation 4.4	29	330	299	30	0	0	330	329	90	0		
5.1	MNH - CNH	Spoof CGA	CGA	Simulation 5.1	74	75	29	1	0	74	105	0	14	0	fail	working
5.2	MNA - CNH	Spoof CGA	CGA	Simulation 5.2	74	75	29	1	0	74	105	0	14	0		
5.3	MNH - CNA	Spoof CGA	CGA	Simulation 5.3	59	60	29	1	0	59	90	0	11	0		
5.4	MNA - CNA	Spoof CGA	CGA	Simulation 5.4	59	60	29	1	0	59	90	0	11	0		
6.1	MNH - CNH	None	RR	Simulation 6.1	74	86	0	12	0	85	75	0	23	0	none	working
6.2	MNA - CNH	None	RR	Simulation 6.2	71	83	0	12	0	82	72	0	22	0		
6.3	MNH - CNA	None	RR	Simulation 6.3	59	69	0	10	0	68	60	0	19	0		
6.4	MNA - CNA	None	RR	Simulation 6.4	57	67	0	10	0	66	58	0	18	0		
7.1	MNH - CNH	HA HoT	RR	Simulation 7.1	173	271	49	51	0	220	221	50	75	0	fail	working
7.2	MNA - CNH	HA HoT	RR	Simulation 7.2	188	290	49	54	0	235	237	50	75	0		
7.3	MNH - CNA	HA HoT	RR	Simulation 7.3	111	203	49	42	0	151	162	50	58	0		
7.4	MNA - CNA	HA HoT	RR	Simulation 7.4	148	246	49	47	0	191	199	50	67	0		
8.1	MNH - CNH	Spoof HA	RR	Simulation 8.1	30	407	309	31	0	1	341	404	90	0	pass	fail
8.2	MNA - CNH	Spoof HA	RR	Simulation 8.2	30	407	310	30	0	2	341	404	88	0		
8.3	MNH - CNA	Spoof HA	RR	Simulation 8.3	29	408	311	30	0	0	342	407	90	0		
8.4	MNA - CNA	Spoof HA	RR	Simulation 8.4	29	408	311	30	0	0	342	407	90	0		
9.1	MNH - CNH	None	DAP	Simulation 9.1	74	86	0	12	0	74	86	0	23	0	none	working
9.2	MNA - CNH	None	DAP	Simulation 9.2	74	86	0	12	0	74	86	0	23	0		
9.3	MNH - CNA	None	DAP	Simulation 9.3	59	69	0	10	0	59	69	0	19	0		
9.4	MNA - CNA	None	DAP	Simulation 9.4	59	69	0	10	0	59	69	0	19	0		
10.1	MNH - CNH	False BU	DAP	Simulation 10.1	74	161	74	37	0	74	186	50	48	0	fail	working
10.2	MNA - CNH	False BU	DAP	Simulation 10.2	74	160	74	38	0	74	185	50	49	0		
10.3	MNH - CNA	False BU	DAP	Simulation 10.3	59	144	74	35	0	59	169	50	44	0		
10.4	MNA - CNA	False BU	DAP	Simulation 10.4	59	144	74	35	0	59	169	50	44	0		
11.1	MNH - CNH	None	DIRR	Simulation 11.1	74	108	0	12	0	107	75	0	23	0	none	working
11.2	MNA - CNH	None	DIRR	Simulation 11.2	71	105	0	12	0	104	72	0	22	0		
11.3	MNH - CNA	None	DIRR	Simulation 11.3	59	87	0	10	0	86	60	0	19	0		
11.4	MNA - CNA	None	DIRR	Simulation 11.4	57	85	0	10	0	84	58	0	18	0		
12.1	MNH - CNH	HA HoT	DIRR	Simulation 12.1	173	371	49	51	0	320	221	50	75	0	fail	working
12.2	MNA - CNH	HA HoT	DIRR	Simulation 12.2	286	547	49	71	0	491	334	50	83	0		
12.3	MNH - CNA	HA HoT	DIRR	Simulation 12.3	111	285	49	42	0	233	162	50	58	0		
12.4	MNA - CNA	HA HoT	DIRR	Simulation 12.4	167	364	49	50	0	310	217	50	73	0		
13.1	MNH - CNH	Spoof HA	DIRR	Simulation 13.1	173	371	49	26	0	295	221	75	50	0	fail	working
13.2	MNA - CNH	Spoof HA	DIRR	Simulation 13.2	137	321	49	21	0	244	186	75	39	0		
13.3	MNH - CNA	Spoof HA	DIRR	Simulation 13.3	111	285	49	17	0	208	162	75	33	0		
13.4	MNA - CNA	Spoof HA	DIRR	Simulation 13.4	153	343	49	23	0	267	202	75	43	0		
14.1	MNH - CNH	None	CGA & RR	Simulation 14.1	74	86	0	12	0	85	75	0	23	0	none	working
14.2	MNA - CNH	None	CGA & RR	Simulation 14.2	71	83	0	12	0	82	72	0	22	0		
14.3	MNH - CNA	None	CGA & RR	Simulation 14.3	59	69	0	10	0	68	60	0	19	0		
14.4	MNA - CNA	None	CGA & RR	Simulation 14.4	57	67	0	10	0	66	58	0	18	0		
15.1	MNH - CNH	Own CGA & HA HoT	CGA & RR	Simulation 15.1	173	271	49	51	0	220	221	50	75	0	fail	working
15.2	MNA - CNH	Own CGA & HA HoT	CGA & RR	Simulation 15.2	188	290	49	54	0	235	237	50	75	0		
15.3	MNH - CNA	Own CGA & HA HoT	CGA & RR	Simulation 15.3	111	203	49	42	0	115	162	50	58	0		
15.4	MNA - CNA	Own CGA & HA HoT	CGA & RR	Simulation 15.4	148	246	49	47	0	191	199	50	67	0		
16.1	MNH - CNH	Spoof CGA & HA HoT	CGA & RR	Simulation 16.1	74	86	29	12	0	85	105	0	23	0	fail	working
16.2	MNA - CNH	Spoof CGA & HA HoT	CGA & RR	Simulation 16.2	71	83	29	12	0	82	102	0	22	0		
16.3	MNH - CNA	Spoof CGA & HA HoT	CGA & RR	Simulation 16.3	59	69	29	10	0	68	90	0	19	0		
16.4	MNA - CNA	Spoof CGA & HA HoT	CGA & RR	Simulation 16.4	57	67	29	10	0	66	88	0	18	0		
17.1	MNH - CNH	Own CGA & Spoof HA	CGA & RR	Simulation 17.1	30	407	309	31	0	1	341	404	90	0	pass	fail
17.2	MNA - CNH	Own CGA & Spoof HA	CGA & RR	Simulation 17.2	30	407	310	30	0	2	341	404	88	0		
17.3	MNH - CNA	Own CGA & Spoof HA	CGA & RR	Simulation 17.3	29	408	311	30	0	0	342	407	90	0		
17.4	MNA - CNA	Own CGA & Spoof HA	CGA & RR	Simulation 17.4	29	408	311	30	0	0	342	407	90	0		
18.1	MNH - CNH	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.1	74	86	29	12	0	85	105	0	23	0	fail	working
18.2	MNA - CNH	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.2	71	83	29	12	0	82	102	0	22	0		
18.3	MNH - CNA	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.3	59	69	29	10	0	68	90	0	19	0		
18.4	MNA - CNA	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.4	57	67	29	10	0	66	88	0	18	0		
19.1	MNH - CNH	None	CGA RR DAP	Simulation 19.1	74	95	0	21	0	84	84	0	30	0	none	working
19.2	MNA - CNH	None	CGA RR DAP	Simulation 19.2	72	90	0	19	0	81	81	0	28	0		
19.3	MNH - CNA	None	CGA RR DAP	Simulation 19.3	59	76	0	17	0	67	68	0	24	0		
19.4	MNA - CNA	None	CGA RR DAP	Simulation 19.4	58	73	0	16	0	66	65	0	23	0		
20.1	MNH - CNH	Own CGA & HA HoT	CGA RR DAP	Simulation 20.1	182	299	49	71	0	226	251	50	92	0	fail	working
20.2	MNA - CNH	Own CGA & HA HoT	CGA RR DAP	Simulation 20.2	197	316	49	73	0	241	266	50	93	0		
20.3	MNH - CNA	Own CGA & HA HoT	CGA RR DAP	Simulation 20.3	119	223	49	56	0	157	183	50	70	0		
20.4	MNA - CNA	Own CGA & HA HoT	CGA RR DAP	Simulation 20.4	152	265	49	63	0	191	220	50	79	0		
21.1	MNH - CNH	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.1	74	95	29	21	0	84	114	0	30	0	fail	working
21.2	MNA - CNH	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.2	72	90	29	19	0	81	111	0	28	0		
21.3	MNH - CNA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.3	59	76	29	17	0	67	98	0	24	0		
21.4	MNA - CNA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.4	58	73	29	16	0	66	95	0	23	0		
22.1	MNH - CNH	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.1	74	211	92	45	0	84	200	92	54	0	fail	working
22.2	MNA - CNH	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.2	72	206	92	43	0	81	197	92	52	0		
22.3	MNH - CNA	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.3	59	192	92	41	0	67	184	92	48	0		
22.4	MNA - CNA	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.4	57	189	92	39	0	64	181	92	46	0		
23.1	MNH - CNH	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.1	74	95	29	21	0	84	114	0	30	0	fail	working
23.2	MNA - CNH	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.2	72	90	29	19	0	81	111	0	28	0		
23.3	MNH - CNA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.3	59	76	29	17	0	67	98	0	24	0		
23.4	MNA - CNA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.4	58	73	29	16	0	66	95	0	23	0		

Table 2. Simulation Results of Packets Sent/Received

24.1	MNH - CNH	None	CGA DIRR DAP	Simulation 24.1	74	115	0	21	0	104	84	0	30	0	none	working
24.2	MNA - CNH	None	CGA DIRR DAP	Simulation 24.2	72	108	0	19	0	99	81	0	28	0		
24.3	MNH - CNA	None	CGA DIRR DAP	Simulation 24.3	59	92	0	17	0	83	68	0	24	0		
24.4	MNA - CNA	None	CGA DIRR DAP	Simulation 24.4	58	89	0	16	0	82	65	0	23	0		
25.1	MNH - CNH	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.1	182	395	49	71	0	322	251	50	92	0	fail	working
25.2	MNA - CNH	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.2	204	426	49	74	0	351	273	50	95	0		
25.3	MNH - CNA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.3	119	303	49	56	0	237	183	50	70	0		
25.4	MNA - CNA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.4	166	372	49	64	0	297	230	50	80	0		
26.1	MNH - CNH	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.1	74	115	29	21	0	104	114	0	30	0	fail	working
26.2	MNA - CNH	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.2	72	108	29	19	0	99	111	0	28	0		
26.3	MNH - CNA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.3	59	92	29	17	0	83	98	0	24	0		
26.4	MNA - CNA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.4	58	89	29	16	0	82	95	0	23	0		
27.1	MNH - CNH	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.1	182	395	49	46	0	297	251	75	67	0	fail	working
27.2	MNA - CNH	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.2	141	336	49	36	0	243	207	75	53	0		
27.3	MNH - CNA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.3	119	303	49	31	0	212	183	75	45	0		
27.4	MNA - CNA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.4	195	418	49	45	0	317	262	75	62	0		
28.1	MNH - CNH	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.1	74	115	29	21	0	104	114	0	30	0	fail	working
28.2	MNA - CNH	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.2	72	108	29	19	0	99	111	0	28	0		
28.3	MNH - CNA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.3	59	92	29	17	0	83	98	0	24	0		
28.4	MNA - CNA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.4	58	89	29	16	0	82	95	0	23	0		
29.1	MNA - CNH - MHA	None	MHA	Simulation 29.1	74	75	0	0	1	74	75	0	0	14	none	working
29.2	MNA - CNA - MHA	None	MHA	Simulation 29.2	59	60	0	0	1	59	60	0	0	11		
30.1	MNA - CNH - MHA	False BU	MHA	Simulation 30.1	74	375	299	0	1	74	375	299	50	14	partial	working
30.2	MNA - CNA - MHA	False BU	MHA	Simulation 30.2	59	360	299	0	1	59	360	299	50	11		
31.1	MNA - CNH - MHA	None	CGA	Simulation 31.1	74	75	0	0	1	74	75	0	0	14	none	working
31.2	MNA - CNA - MHA	None	CGA	Simulation 31.2	59	60	0	0	1	59	60	0	0	11		
32.1	MNA - CNH - MHA	Own CGA	CGA	Simulation 32.1	74	375	299	0	1	74	375	299	50	14	partial	working
32.2	MNA - CNA - MHA	Own CGA	CGA	Simulation 32.2	59	360	299	0	1	59	360	299	50	11		
33.1	MNA - CNH - MHA	Spoof CGA	CGA	Simulation 33.1	74	75	29	0	1	74	105	0	0	14	fail	working
33.2	MNA - CNA - MHA	Spoof CGA	CGA	Simulation 33.2	59	60	29	0	1	59	90	0	0	11		
34.1	MNA - CNH - MHA	None	RR	Simulation 34.1	74	86	0	0	12	85	75	0	0	23	none	working
34.2	MNA - CNA - MHA	None	RR	Simulation 34.2	59	69	0	0	10	68	60	0	0	19		
35.1	MNA - CNH - MHA	HA HoT	RR	Simulation 35.1	173	271	49	0	51	220	221	50	0	75	fail	working
35.2	MNA - CNA - MHA	HA HoT	RR	Simulation 35.2	111	203	49	0	42	151	162	50	0	58		
36.1	MNA - CNH - MHA	Spoof HA	RR	Simulation 36.1	74	429	299	0	12	85	375	342	43	23	partial	working
36.2	MNA - CNA - MHA	Spoof HA	RR	Simulation 36.2	59	421	299	0	10	68	360	342	43	19		
37.1	MNA - CNH - MHA	None	DAP	Simulation 37.1	74	86	0	0	12	74	86	0	0	23	none	working
37.2	MNA - CNA - MHA	None	DAP	Simulation 37.2	59	69	0	0	10	59	69	0	0	19		
38.1	MNA - CNH - MHA	False BU	DAP	Simulation 38.1	74	161	74	0	37	74	168	50	0	48	fail	working
38.2	MNA - CNA - MHA	False BU	DAP	Simulation 38.2	59	144	74	0	36	59	169	50	0	45		
39.1	MNA - CNH - MHA	None	DIRR	Simulation 39.1	74	108	0	0	23	107	75	0	0	34	none	working
39.2	MNA - CNA - MHA	None	DIRR	Simulation 39.2	59	87	0	0	19	86	60	0	0	28		
40.1	MNA - CNH - MHA	HA HoT	DIRR	Simulation 40.1	173	371	49	0	101	320	221	50	0	125	fail	working
40.2	MNA - CNA - MHA	HA HoT	DIRR	Simulation 40.2	111	285	49	0	83	233	162	50	0	99		
41.1	MNA - CNH - MHA	Spoof HA	DIRR	Simulation 41.1	173	373	49	0	76	295	221	75	0	100	fail	working
41.2	MNA - CNA - MHA	Spoof HA	DIRR	Simulation 41.2	111	285	49	0	58	208	162	75	0	74		
42.1	MNA - CNH - MHA	None	CGA & RR	Simulation 42.1	74	86	0	0	12	85	75	0	0	23	none	working
42.2	MNA - CNA - MHA	None	CGA & RR	Simulation 42.2	59	69	0	0	10	68	60	0	0	19		
43.1	MNA - CNH - MHA	Own CGA & HA HoT	CGA & RR	Simulation 43.1	173	271	49	0	51	220	221	50	0	75	fail	working
43.2	MNA - CNA - MHA	Own CGA & HA HoT	CGA & RR	Simulation 43.2	111	203	49	0	42	151	162	50	0	58		
44.1	MNA - CNH - MHA	Spoof CGA & HA HoT	CGA & RR	Simulation 44.1	74	86	29	0	12	85	105	0	0	23	fail	working
44.2	MNA - CNA - MHA	Spoof CGA & HA HoT	CGA & RR	Simulation 44.2	59	69	29	0	10	68	90	0	0	19		
45.1	MNA - CNH - MHA	Own CGA & Spoof HA	CGA & RR	Simulation 45.1	74	429	299	0	12	85	375	342	43	23	partial	working
45.2	MNA - CNA - MHA	Own CGA & Spoof HA	CGA & RR	Simulation 45.2	59	412	299	0	10	68	360	342	43	19		
46.1	MNA - CNH - MHA	Spoof CGA & Spoof HA	CGA & RR	Simulation 46.1	74	86	29	0	12	85	105	0	0	23	fail	working
46.2	MNA - CNA - MHA	Spoof CGA & Spoof HA	CGA & RR	Simulation 46.2	59	69	29	0	10	68	90	0	0	19		
47.1	MNA - CNH - MHA	None	CGA RR DAP	Simulation 47.1	74	95	0	0	21	84	84	0	0	30	none	working
47.2	MNA - CNA - MHA	None	CGA RR DAP	Simulation 47.2	59	76	0	0	17	67	68	0	0	24		
48.1	MNA - CNH - MHA	Own CGA & HA HoT	CGA RR DAP	Simulation 48.1	182	299	49	0	71	226	250	50	0	92	fail	working
48.2	MNA - CNA - MHA	Own CGA & HA HoT	CGA RR DAP	Simulation 48.2	119	223	49	0	56	157	182	50	0	70		
49.1	MNA - CNH - MHA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 49.1	74	95	29	0	21	84	114	0	0	30	fail	working
49.2	MNA - CNA - MHA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 49.2	59	76	29	0	17	67	98	0	0	24		
50.1	MNA - CNH - MHA	Own CGA & Spoof HA	CGA RR DAP	Simulation 50.1	74	211	92	0	45	84	200	92	0	54	fail	working
50.2	MNA - CNA - MHA	Own CGA & Spoof HA	CGA RR DAP	Simulation 50.2	59	192	92	0	41	67	184	92	0	48		
51.1	MNA - CNH - MHA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 51.1	74	95	29	0	21	84	114	0	0	30	fail	working
51.2	MNA - CNA - MHA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 51.2	59	76	29	0	17	67	98	0	0	24		
52.1	MNA - CNH - MHA	None	CGA DIRR DAP	Simulation 52.1	74	115	0	0	31	104	84	0	0	40	none	working
52.2	MNA - CNA - MHA	None	CGA DIRR DAP	Simulation 52.2	59	92	0	0	25	83	68	0	0	32		
53.1	MNA - CNH - MHA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 53.1	182	395	49	0	119	322	250	50	0	140	fail	working
53.2	MNA - CNA - MHA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 53.2	119	303	49	0	96	237	182	50	0	110		
54.1	MNA - CNH - MHA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 54.1	74	115	29	0	31	104	114	0	0	40	fail	working
54.2	MNA - CNA - MHA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 54.2	59	92	29	0	25	83	98	0	0	32		
55.1	MNA - CNH - MHA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 55.1	182	395	49	0	94	297	250	75	0	115	fail	working
55.2	MNA - CNA - MHA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 55.2	119	303	49	0	71	212	182	75	0	85		
56.1	MNA - CNH - MHA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 56.1	74	115	29	0	31	104	114	0	0	40	fail	working
56.2	MNA - CNA - MHA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 56.2	59	92	29	0	25	83	98	0	0	32		

Table 3. Simulation Results of Packets Sent/Received

Simulation results Packets Min/Max Hop Count in 60 Seconds

Network config:
MNH - CNH= Mobile Node at home to Correspondent Node at home
MNA - CNH= Mobile Node at Away from home to Correspondent Node at home
MNH - CNA= Mobile Node at home to Correspondent Node away
MNA - CNA= Mobile Node at Away from home to Correspondent Node away

Nodes:
MN = Mobile Node
CN = Corresponding Node
ATK = Attacker
HA = Home Agent
MHA = Mobile Home Agent

					Hop Count Min					Hop Count Max				
Sim. No.	Network Config	Attack type	Security Type		MN	CN	ATK	HA	MHA	MN	CN	ATK	HA	MHA
1.1	MNH - CNH	None	0	Simulation 1.1	4	4	0	2	0	4	4	0	2	0
1.2	MNA - CNH	None	0	Simulation 1.2	4	4	0	2	0	6	4	0	4	0
1.3	MNH - CNA	None	0	Simulation 1.3	5	5	0	2	0	5	5	0	3	0
1.4	MNA - CNA	None	0	Simulation 1.4	5	5	0	3	0	7	5	0	4	0
2.1	MNH - CNH	False BU	0	Simulation 2.1	0	1	1	1	0	0	4	4	2	0
2.2	MNA - CNH	False BU	0	Simulation 2.2	0	1	1	1	0	0	4	4	2	0
2.3	MNH - CNA	False BU	0	Simulation 2.3	0	1	1	1	0	0	5	5	3	0
2.4	MNA - CNA	False BU	0	Simulation 2.4	0	1	1	1	0	0	5	5	3	0
3.1	MNH - CNH	None	CGA	Simulation 3.1	4	4	0	2	0	4	4	0	2	0
3.2	MNA - CNH	None	CGA	Simulation 3.2	4	4	0	2	0	6	4	0	4	0
3.3	MNH - CNA	None	CGA	Simulation 3.3	5	5	0	2	0	5	5	0	3	0
3.4	MNA - CNA	None	CGA	Simulation 3.4	5	5	0	3	0	7	5	0	4	0
4.1	MNH - CNH	Own CGA	CGA	Simulation 4.1	0	1	1	1	0	0	4	4	2	0
4.2	MNA - CNH	Own CGA	CGA	Simulation 4.2	0	1	1	1	0	0	4	4	2	0
4.3	MNH - CNA	Own CGA	CGA	Simulation 4.3	0	1	1	1	0	0	5	5	3	0
4.4	MNA - CNA	Own CGA	CGA	Simulation 4.4	0	1	1	1	0	0	5	5	3	0
5.1	MNH - CNH	Spoof CGA	CGA	Simulation 5.1	4	1	0	2	0	4	4	0	2	0
5.2	MNA - CNH	Spoof CGA	CGA	Simulation 5.2	4	1	0	2	0	6	4	0	4	0
5.3	MNH - CNA	Spoof CGA	CGA	Simulation 5.3	5	1	0	2	0	5	5	0	3	0
5.4	MNA - CNA	Spoof CGA	CGA	Simulation 5.4	5	1	0	3	0	7	5	0	4	0
6.1	MNH - CNH	None	RR	Simulation 6.1	4	4	0	2	0	4	4	0	2	0
6.2	MNA - CNH	None	RR	Simulation 6.2	4	4	0	2	0	6	4	0	4	0
6.3	MNH - CNA	None	RR	Simulation 6.3	5	5	0	2	0	5	5	0	3	0
6.4	MNA - CNA	None	RR	Simulation 6.4	5	5	0	3	0	7	5	0	4	0
7.1	MNH - CNH	HA HoT	RR	Simulation 7.1	4	1	1	2	0	4	4	1	2	0
7.2	MNA - CNH	HA HoT	RR	Simulation 7.2	4	1	1	2	0	6	4	1	4	0
7.3	MNH - CNA	HA HoT	RR	Simulation 7.3	5	1	1	2	0	5	5	1	3	0
7.4	MNA - CNA	HA HoT	RR	Simulation 7.4	5	1	1	3	0	7	5	1	4	0
8.1	MNH - CNH	Spoof HA	RR	Simulation 8.1	4	1	1	1	0	4	4	4	2	0
8.2	MNA - CNH	Spoof HA	RR	Simulation 8.2	4	1	1	1	0	6	4	4	2	0
8.3	MNH - CNA	Spoof HA	RR	Simulation 8.3	0	1	1	1	0	0	5	5	3	0
8.4	MNA - CNA	Spoof HA	RR	Simulation 8.4	0	1	1	1	0	0	5	5	3	0
9.1	MNH - CNH	None	DAP	Simulation 9.1	4	4	0	2	0	4	4	0	2	0
9.2	MNA - CNH	None	DAP	Simulation 9.2	4	4	0	2	0	6	4	0	4	0
9.3	MNH - CNA	None	DAP	Simulation 9.3	5	5	0	2	0	5	6	0	3	0
9.4	MNA - CNA	None	DAP	Simulation 9.4	5	5	0	3	0	7	6	0	4	0
10.1	MNH - CNH	False BU	DAP	Simulation 10.1	4	1	1	2	0	4	4	1	2	0
10.2	MNA - CNH	False BU	DAP	Simulation 10.2	4	1	1	2	0	4	4	1	4	0
10.3	MNH - CNA	False BU	DAP	Simulation 10.3	5	1	1	2	0	5	6	1	3	0
10.4	MNA - CNA	False BU	DAP	Simulation 10.4	5	1	1	3	0	7	6	1	4	0
11.1	MNH - CNH	None	DIRR	Simulation 11.1	4	4	0	2	0	4	4	0	2	0
11.2	MNA - CNH	None	DIRR	Simulation 11.2	4	4	0	2	0	6	4	0	4	0
11.3	MNH - CNA	None	DIRR	Simulation 11.3	5	5	0	2	0	5	5	0	3	0
11.4	MNA - CNA	None	DIRR	Simulation 11.4	5	5	0	3	0	7	5	0	4	0
12.1	MNH - CNH	HA HoT	DIRR	Simulation 12.1	4	1	1	2	0	4	4	1	2	0
12.2	MNA - CNH	HA HoT	DIRR	Simulation 12.2	4	1	1	2	0	6	4	1	4	0
12.3	MNH - CNA	HA HoT	DIRR	Simulation 12.3	5	1	1	2	0	5	5	1	3	0
12.4	MNA - CNA	HA HoT	DIRR	Simulation 12.4	5	1	1	3	0	7	5	1	4	0
13.1	MNH - CNH	Spoof HA	DIRR	Simulation 13.1	4	1	1	2	0	4	4	1	2	0
13.2	MNA - CNH	Spoof HA	DIRR	Simulation 13.2	4	1	1	2	0	6	4	1	4	0
13.3	MNH - CNA	Spoof HA	DIRR	Simulation 13.3	5	1	1	2	0	5	5	1	3	0
13.4	MNA - CNA	Spoof HA	DIRR	Simulation 13.4	5	1	1	3	0	7	5	1	4	0
14.1	MNH - CNH	None	CGA & RR	Simulation 14.1	4	4	0	2	0	4	4	0	2	0
14.2	MNA - CNH	None	CGA & RR	Simulation 14.2	4	4	0	2	0	6	4	0	4	0
14.3	MNH - CNA	None	CGA & RR	Simulation 14.3	5	5	0	2	0	5	5	0	3	0
14.4	MNA - CNA	None	CGA & RR	Simulation 14.4	5	5	0	3	0	7	5	0	4	0
15.1	MNH - CNH	Own CGA & HA HoT	CGA & RR	Simulation 15.1	4	1	1	2	0	4	4	1	2	0
15.2	MNA - CNH	Own CGA & HA HoT	CGA & RR	Simulation 15.2	4	1	1	2	0	6	4	1	4	0
15.3	MNH - CNA	Own CGA & HA HoT	CGA & RR	Simulation 15.3	5	1	1	2	0	5	5	1	3	0
15.4	MNA - CNA	Own CGA & HA HoT	CGA & RR	Simulation 15.4	5	1	1	3	0	7	5	1	4	0
16.1	MNH - CNH	Spoof CGA & HA HoT	CGA & RR	Simulation 16.1	4	1	0	2	0	4	4	0	2	0
16.2	MNA - CNH	Spoof CGA & HA HoT	CGA & RR	Simulation 16.2	4	1	0	2	0	6	4	0	4	0
16.3	MNH - CNA	Spoof CGA & HA HoT	CGA & RR	Simulation 16.3	5	1	0	2	0	5	5	0	3	0
16.4	MNA - CNA	Spoof CGA & HA HoT	CGA & RR	Simulation 16.4	5	1	0	3	0	7	5	0	4	0
17.1	MNH - CNH	Own CGA & Spoof HA	CGA & RR	Simulation 17.1	4	1	1	1	0	4	4	4	2	0
17.2	MNA - CNH	Own CGA & Spoof HA	CGA & RR	Simulation 17.2	4	1	1	1	0	6	4	4	2	0
17.3	MNH - CNA	Own CGA & Spoof HA	CGA & RR	Simulation 17.3	0	1	1	1	0	0	5	5	3	0
17.4	MNA - CNA	Own CGA & Spoof HA	CGA & RR	Simulation 17.4	0	1	1	1	0	0	5	5	3	0
18.1	MNH - CNH	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.1	4	1	0	2	0	4	4	0	2	0
18.2	MNA - CNH	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.2	4	1	0	2	0	6	4	0	4	0
18.3	MNH - CNA	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.3	5	1	0	2	0	5	5	0	3	0
18.4	MNA - CNA	Spoof CGA & Spoof HA	CGA & RR	Simulation 18.4	5	1	0	3	0	7	5	0	4	0
19.1	MNH - CNH	None	CGA RR DAP	Simulation 19.1	4	4	0	2	0	4	4	0	2	0
19.2	MNA - CNH	None	CGA RR DAP	Simulation 19.2	4	4	0	2	0	6	4	0	4	0
19.3	MNH - CNA	None	CGA RR DAP	Simulation 19.3	5	5	0	2	0	5	6	0	3	0
19.4	MNA - CNA	None	CGA RR DAP	Simulation 19.4	5	5	0	3	0	7	6	0	4	0
20.1	MNH - CNH	Own CGA & HA HoT	CGA RR DAP	Simulation 20.1	4	1	1	2	0	4	4	1	2	0
20.2	MNA - CNH	Own CGA & HA HoT	CGA RR DAP	Simulation 20.2	4	1	1	2	0	6	4	1	4	0
20.3	MNH - CNA	Own CGA & HA HoT	CGA RR DAP	Simulation 20.3	5	1	1	2	0	5	6	1	3	0
20.4	MNA - CNA	Own CGA & HA HoT	CGA RR DAP	Simulation 20.4	5	1	1	3	0	7	6	1	4	0
21.1	MNH - CNH	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.1	4	1	0	2	0	4	4	0	2	0
21.2	MNA - CNH	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.2	4	1	0	2	0	6	4	0	4	0
21.3	MNH - CNA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.3	5	1	0	2	0	5	6	0	3	0
21.4	MNA - CNA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 21.4	5	1	0	3	0	7	6	0	4	0
22.1	MNH - CNH	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.1	4	1	1	2	0	4	4	1	2	0
22.2	MNA - CNH	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.2	4	1	1	2	0	6	4	1	4	0
22.3	MNH - CNA	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.3	5	1	1	2	0	5	6	1	3	0
22.4	MNA - CNA	Own CGA & Spoof HA	CGA RR DAP	Simulation 22.4	5	1	1	3	0	7	6	1	4	0
23.1	MNH - CNH	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.1	4	1	0	2	0	4	4	0	2	0
23.2	MNA - CNH	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.2	4	1	0	2	0	6	4	0	4	0
23.3	MNH - CNA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.3	5	1	0	2	0	5	6	0	3	0
23.4	MNA - CNA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 23.4	5	1	0	3	0	7	6	0	4	0

Table 4. Simulation Results of Min/Max Hop Count

24.1	MNH - CNH	None	CGA DIRR DAP	Simulation 24.1	4	4	0	2	0	4	4	0	2	0
24.2	MNA - CNH	None	CGA DIRR DAP	Simulation 24.2	4	4	0	2	0	6	4	0	4	0
24.3	MNH - CNA	None	CGA DIRR DAP	Simulation 24.3	5	5	0	2	0	5	6	0	3	0
24.4	MNA - CNA	None	CGA DIRR DAP	Simulation 24.4	5	5	0	3	0	7	6	0	4	0
25.1	MNH - CNH	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.1	4	1	1	2	0	4	4	1	2	0
25.2	MNA - CNH	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.2	4	1	1	2	0	6	4	1	4	0
25.3	MNH - CNA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.3	5	1	1	2	0	5	6	1	3	0
25.4	MNA - CNA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 25.4	5	1	1	3	0	7	6	1	4	0
26.1	MNH - CNH	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.1	4	1	0	2	0	4	4	0	2	4
26.2	MNA - CNH	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.2	4	1	0	2	0	6	4	0	4	0
26.3	MNH - CNA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.3	5	1	0	2	0	5	6	0	3	0
26.4	MNA - CNA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 26.4	5	1	0	3	0	7	6	0	4	0
27.1	MNH - CNH	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.1	4	1	1	2	0	4	4	1	2	0
27.2	MNA - CNH	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.2	4	1	1	2	0	6	4	1	4	0
27.3	MNH - CNA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.3	5	1	1	2	0	5	6	1	3	0
27.4	MNA - CNA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 27.4	5	1	1	3	0	7	6	1	4	0
28.1	MNH - CNH	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.1	4	1	0	2	0	4	4	0	2	0
28.2	MNA - CNH	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.2	4	1	0	2	0	6	4	0	4	0
28.3	MNH - CNA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.3	5	1	0	2	0	5	6	0	3	0
28.4	MNA - CNA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 28.4	5	1	0	3	0	7	6	0	4	0
29.1	MNA - CNH - MHA	None	MHA	Simulation 29.1	4	4	0	0	1	4	4	0	0	3
29.2	MNA - CNA - MHA	None	MHA	Simulation 29.2	5	5	0	0	1	5	5	0	0	4
30.1	MNA - CNH - MHA	False BU	MHA	Simulation 30.1	4	1	1	1	1	4	4	1	1	3
30.2	MNA - CNA - MHA	False BU	MHA	Simulation 30.2	5	1	1	1	1	5	5	1	1	4
31.1	MNA - CNH - MHA	None	CGA	Simulation 31.1	4	4	0	0	1	4	4	0	0	3
31.2	MNA - CNA - MHA	None	CGA	Simulation 31.2	5	5	0	0	1	5	5	0	0	4
32.1	MNA - CNH - MHA	Own CGA	CGA	Simulation 32.1	4	1	1	1	1	4	4	1	1	3
32.2	MNA - CNA - MHA	Own CGA	CGA	Simulation 32.2	5	1	1	1	1	5	5	1	1	4
33.1	MNA - CNH - MHA	Spoof CGA	CGA	Simulation 33.1	4	1	0	0	1	4	4	0	0	3
33.2	MNA - CNA - MHA	Spoof CGA	CGA	Simulation 33.2	5	1	0	0	1	5	5	0	0	4
34.1	MNA - CNH - MHA	None	RR	Simulation 34.1	4	4	0	0	1	4	4	0	0	3
34.2	MNA - CNA - MHA	None	RR	Simulation 34.2	5	5	0	0	1	5	5	0	0	4
35.1	MNA - CNH - MHA	HA HoT	RR	Simulation 35.1	4	1	1	0	1	4	4	1	0	3
35.2	MNA - CNA - MHA	HA HoT	RR	Simulation 35.2	5	1	1	0	1	5	5	1	0	4
36.1	MNA - CNH - MHA	Spoof HA	RR	Simulation 36.1	4	1	1	1	1	4	4	1	1	3
36.2	MNA - CNA - MHA	Spoof HA	RR	Simulation 36.2	5	1	1	1	1	5	5	1	1	4
37.1	MNA - CNH - MHA	None	DAP	Simulation 37.1	4	4	0	0	1	4	6	0	0	3
37.2	MNA - CNA - MHA	None	DAP	Simulation 37.2	5	5	0	0	1	5	8	0	0	4
38.1	MNA - CNH - MHA	False BU	DAP	Simulation 38.1	4	1	1	0	1	4	6	1	0	3
38.2	MNA - CNA - MHA	False BU	DAP	Simulation 38.2	5	1	1	0	1	5	8	1	0	4
39.1	MNA - CNH - MHA	None	DIRR	Simulation 39.1	4	4	0	0	1	4	4	0	0	3
39.2	MNA - CNA - MHA	None	DIRR	Simulation 39.2	5	5	0	0	1	5	5	0	0	4
40.1	MNA - CNH - MHA	HA HoT	DIRR	Simulation 40.1	4	1	1	0	1	4	4	1	0	3
40.2	MNA - CNA - MHA	HA HoT	DIRR	Simulation 40.2	5	1	1	0	1	5	5	1	0	4
41.1	MNA - CNH - MHA	Spoof HA	DIRR	Simulation 41.1	4	1	1	0	1	4	4	1	0	3
41.2	MNA - CNA - MHA	Spoof HA	DIRR	Simulation 41.2	5	1	1	0	1	5	5	1	0	4
42.1	MNA - CNH - MHA	None	CGA & RR	Simulation 42.1	4	4	0	0	1	4	4	0	0	3
42.2	MNA - CNA - MHA	None	CGA & RR	Simulation 42.2	5	5	0	0	1	5	5	0	0	4
43.1	MNA - CNH - MHA	Own CGA & HA HoT	CGA & RR	Simulation 43.1	4	1	1	0	1	4	4	1	0	3
43.2	MNA - CNA - MHA	Own CGA & HA HoT	CGA & RR	Simulation 43.2	5	1	1	0	1	5	5	1	0	4
44.1	MNA - CNH - MHA	Spoof CGA & HA HoT	CGA & RR	Simulation 44.1	4	1	0	0	1	4	4	0	0	3
44.2	MNA - CNA - MHA	Spoof CGA & HA HoT	CGA & RR	Simulation 44.2	5	1	0	0	1	5	5	0	0	4
45.1	MNA - CNH - MHA	Own CGA & Spoof HA	CGA & RR	Simulation 45.1	4	1	1	1	1	4	4	1	1	3
45.2	MNA - CNA - MHA	Own CGA & Spoof HA	CGA & RR	Simulation 45.2	5	1	1	1	1	5	5	1	1	4
46.1	MNA - CNH - MHA	Spoof CGA & Spoof HA	CGA & RR	Simulation 46.1	4	1	0	0	1	4	4	0	0	3
46.2	MNA - CNA - MHA	Spoof CGA & Spoof HA	CGA & RR	Simulation 46.2	5	1	0	0	1	5	5	0	0	4
47.1	MNA - CNH - MHA	None	CGA RR DAP	Simulation 47.1	4	4	0	0	1	4	6	0	0	3
47.2	MNA - CNA - MHA	None	CGA RR DAP	Simulation 47.2	5	5	0	0	1	5	8	0	0	4
48.1	MNA - CNH - MHA	Own CGA & HA HoT	CGA RR DAP	Simulation 48.1	4	1	1	0	1	4	6	1	0	3
48.2	MNA - CNA - MHA	Own CGA & HA HoT	CGA RR DAP	Simulation 48.2	5	1	1	0	1	5	8	1	0	4
49.1	MNA - CNH - MHA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 49.1	4	1	0	0	1	4	6	0	0	3
49.2	MNA - CNA - MHA	Spoof CGA & HA HoT	CGA RR DAP	Simulation 49.2	5	1	0	0	1	5	8	0	0	4
50.1	MNA - CNH - MHA	Own CGA & Spoof HA	CGA RR DAP	Simulation 50.1	4	1	1	0	1	4	6	1	0	3
50.2	MNA - CNA - MHA	Own CGA & Spoof HA	CGA RR DAP	Simulation 50.2	5	1	1	0	1	5	8	1	0	4
51.1	MNA - CNH - MHA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 51.1	4	1	0	0	1	4	6	0	0	3
51.2	MNA - CNA - MHA	Spoof CGA & Spoof HA	CGA RR DAP	Simulation 51.2	5	1	0	0	1	5	8	0	0	4
52.1	MNA - CNH - MHA	None	CGA DIRR DAP	Simulation 52.1	4	4	0	0	1	4	6	0	0	3
52.2	MNA - CNA - MHA	None	CGA DIRR DAP	Simulation 52.2	5	5	0	0	1	5	8	0	0	4
53.1	MNA - CNH - MHA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 53.1	4	1	0	0	1	4	6	0	0	3
53.2	MNA - CNA - MHA	Own CGA & HA HoT	CGA DIRR DAP	Simulation 53.2	5	1	1	0	1	5	8	1	0	4
54.1	MNA - CNH - MHA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 54.1	4	1	0	0	1	4	6	0	0	3
54.2	MNA - CNA - MHA	Spoof CGA & HA HoT	CGA DIRR DAP	Simulation 54.2	5	1	0	0	1	5	8	0	0	4
55.1	MNA - CNH - MHA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 55.1	4	1	1	0	1	4	6	1	0	3
55.2	MNA - CNA - MHA	Own CGA & Spoof HA	CGA DIRR DAP	Simulation 55.2	5	1	1	0	1	5	8	1	0	4
56.1	MNA - CNH - MHA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 56.1	4	1	0	0	1	4	6	0	0	3
56.2	MNA - CNA - MHA	Spoof CGA & Spoof HA	CGA DIRR DAP	Simulation 56.2	5	1	0	0	1	5	8	0	0	4

Table 5. Simulation Results of Min/Max Hop Count

					Packets Sent					Packets Received				
Sim. No.	Network Config	Attack type	Security Type		MN	CN	ATK	HA	HA2	MN	CN	ATK	HA	HA2
1.1	MNH - CNH	None	None	Simulation 1.1	74	75	0	1	0	74	75	0	14	0
6.1	MNH - CNH	None	RR	Simulation 6.1	74	86	0	12	0	85	75	0	23	0
11.1	MNH - CNH	None	DIRR	Simulation 11.1	74	108	0	12	11	107	75	0	23	11

Table 6. Simulation Results Showing Packet Data on Second Network

6.6 Analysis of Results

The first simulation was run to gather data on how the network operated under normal circumstances with communication between the mobile node and the correspondent only with no attacks or security. This gives us the base line control reading that the rest of the network simulations will be compared to. Attack simulations were then performed using a false binding update and the results compared to the control.

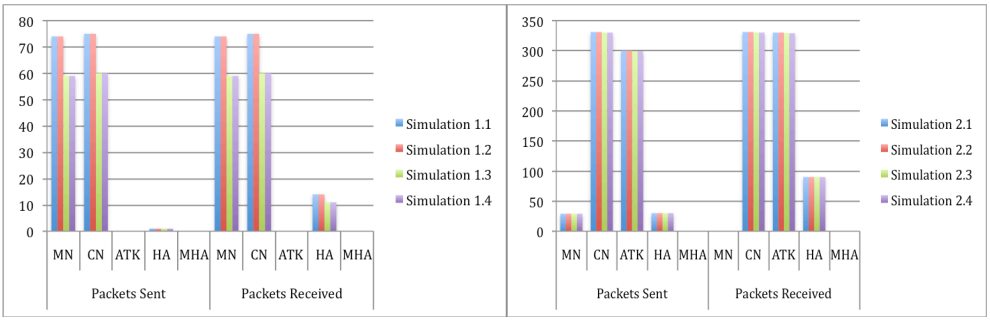

Figure 30. Comparison of Control Simulation 1 with Attack Simulation 2

Each simulation was run four times with a different network configurations to test the effectiveness of the security on the attacks in different situations.
The Configurations are

- Mobile Node at home communicating with static correspondent.
- Mobile Node on a foreign network to a static correspondent.
- Mobile Node at home communicating with mobile correspondent.
- Mobile Node on a foreign network to a mobile correspondent.

Each are run in a simulation, therefore simulation 1.2 will be the first simulation with the second configuration.
Looking at the results of simulation 2 (Figure 30), we can see that the attacker has clearly succeeded in hijacking the communication away from the mobile node as the mobile node has repeatedly sent some packets to restart communication but received none in reply.

You will also notice that the attacker has a much higher rate of communication compared to the Mobile node. This is due to the attacker being one hop away from any node in the network while the mobile node is four to five hops away. This can be seen if we compare the data sent by the mobile node in simulation 1.1 to the data sent by the attacker in simulation 2.1. Mobile Node sent 74 packets and the Attacker sent 300. However divide this number by the amount of hops and you get 75 packets which is comparable.

The simulation software also provides a graphical representation of the communication occurring, which shows visually what the results support. Useful if the results become ambiguous.

The first security solution to be tested was Cryptographically Generated Addresses.
This was first tested in a control environment to determine what impact it has on network communications.

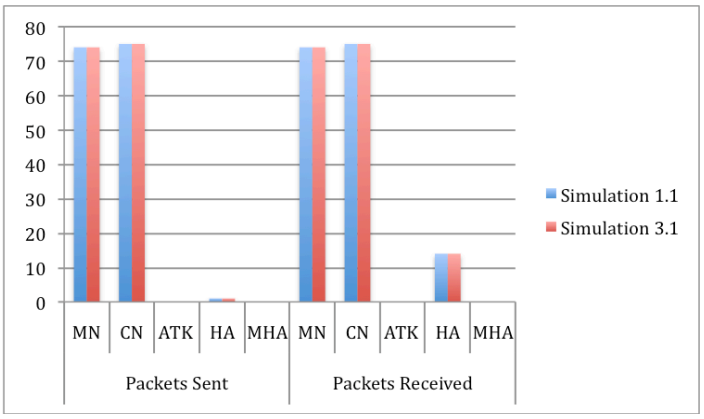

Figure 31. Comparison of Control Simulation 1 with CGA Control Simulation 3

The Figure 31 shows that there is no impact on network communication and the number of packets sent during the use of CGA are identical to a network without it.

The next simulation tested CGA against an attacker using it's own CGA address.
The attacker managed to bypass this security measure as it used it's own cryptographically generated address as this security solution only checks that you own the address not who is using it.
Looking at the data in Figure 32 from simulation 2.1 where the attacker used a false binding update in a non secured network compared to simulation 4.1 which used CGA we can see that the security measure had no impact on this flavour of attack as the numbers are identical.

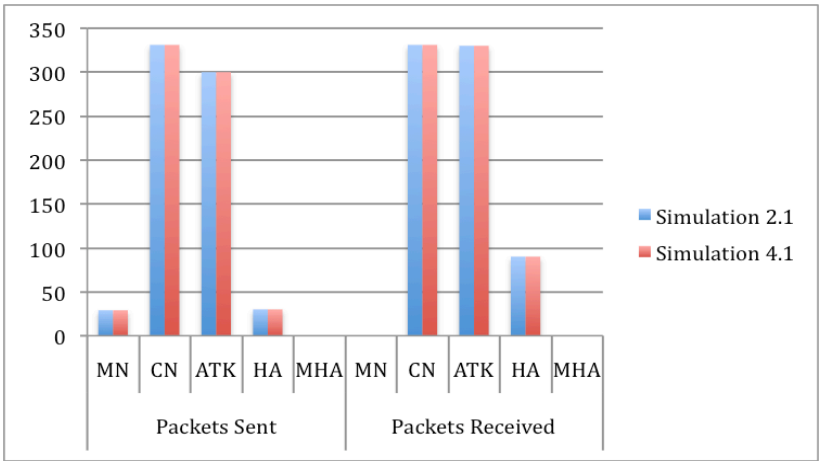

Figure 32. Comparison of Control Attack Simulation 2 with CGA Attack Simulation 4

This is true for all configurations of the network, which can be seen in Figure 33. The only difference being a one-packet discrepancy with simulations that have a larger hop count.

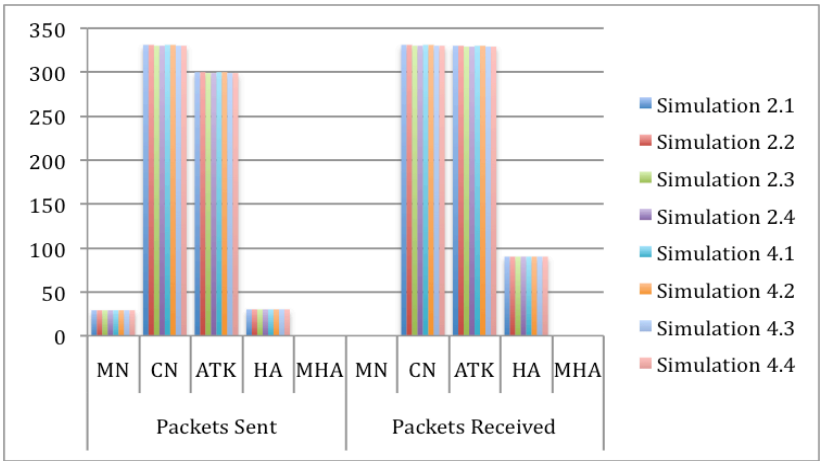

Figure 33. Comparison of Packets Sent/Received in Control Attack Simulation 2 with CGA Attack Simulation 4

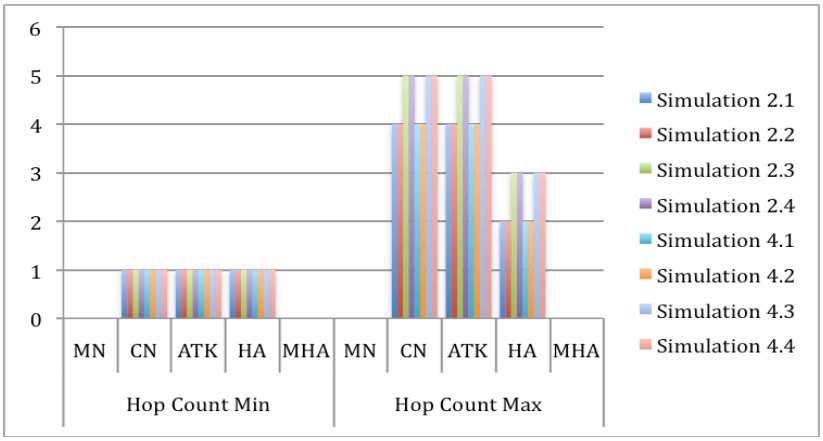

Figure 34. Comparison of Hop Counts in Control Attack Simulation 2 with CGA Attack Simulation 4

As you can see in Figure 34 the min hop count and max hop count are identical for both scenarios.

But how will CGA fare against an attack, which is attempting to spoof the address of another node?

Looking at the graphical animation of the simulation in Figure 35 we can see that the attacker failed to authenticate its packets.

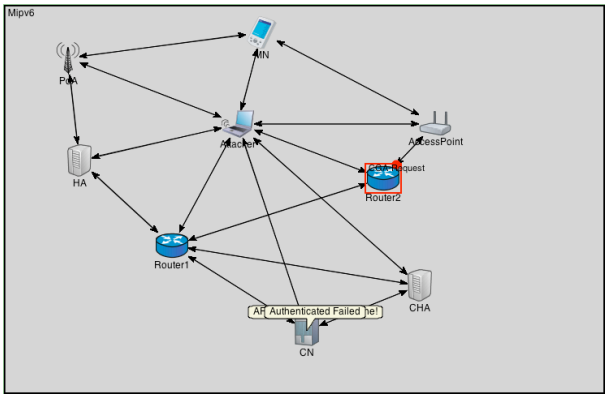

Figure 35. Network Simulation Graphical Feedback

In the simulation shown in Figure 36, the attacker spoofs the address of the Home Agent however is unable to successfully bypass the security. Comparing this with the CGA control network without an attack it can be seen that there has been no effect on the data traffic to and from the Mobile Node but there is an increase in the packets received by the Correspondent Node. This is due to the repeated attempts of the attacker to connect to the correspondent node.

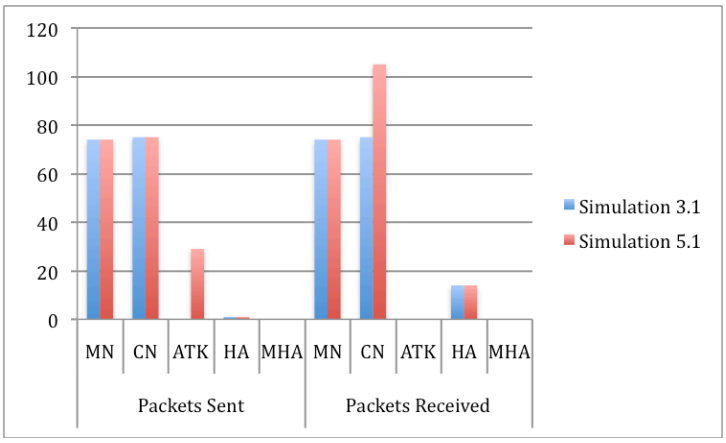

Figure 36. Comparison of Control CGA Simulation 3 with CGA Attack Simulation 5

This shows that Cryptographically Generated Addresses do prevent spoofing of addresses the attacker does not own, however this can be bypassed by using it's its own address.

The next security solution to be tested is Return Routability.

Firstly a control simulation is run to gather a base line reading on how the solution compares to the standard network control which can be seen in Figure 37.

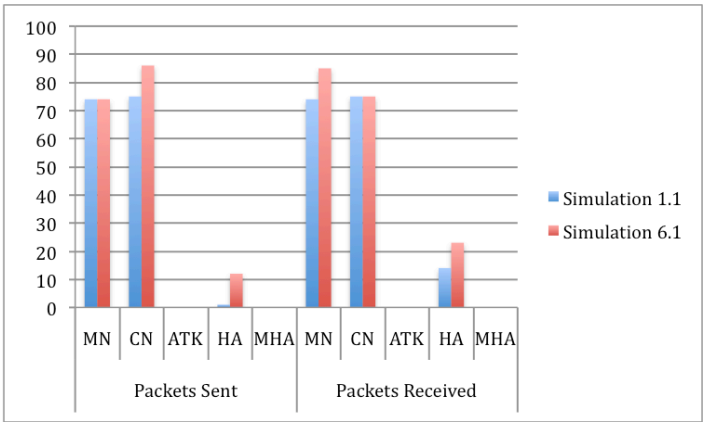

Figure 37. Comparison of Control Simulation 1 with RR Control Simulation 6

It can be seen that there is an increase in the packets sent by the CN (correspondent Node) and received by the Mobile node. This is because the Correspondent generates two security packets, which are sent to the Mobile Node. The first is sent directly to the Mobile Node and the second is sent via the Home Agent, which then forwards the packet to the mobile node. This also explains the increase of data traffic to and from the Home Agent.

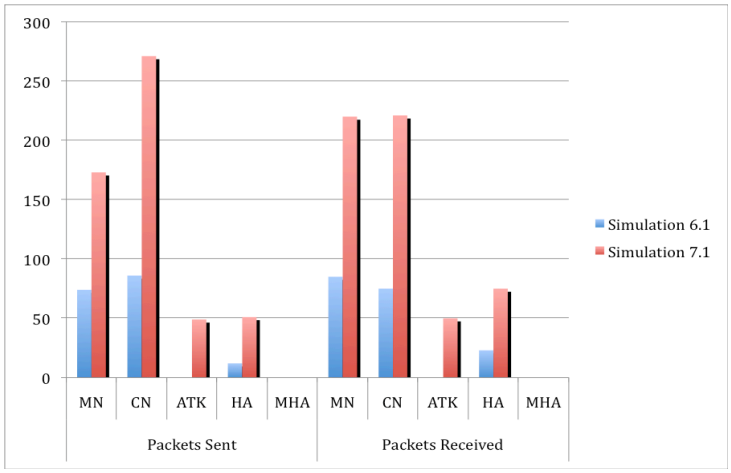

Figure 38. Comparison of RR Control Simulation 6 with RR Attack Simulation 7

We can see in Figure 38 that there is an increase in data traffic by two to three times. This is due not only to the MN node receiving two security tokens from the CN but when the Attacker requests the tokens only one of them are sent to it. The other is sent to the MN via the HA.

As the Attacker is unable to gain both tokens it is unable to create a binding key to continue the communication with the Correspondent Node. Therefore this solution is secure in this scenario.

However this is not the case in the next simulation when the Attacker spoofs the address of the Home Agent too.

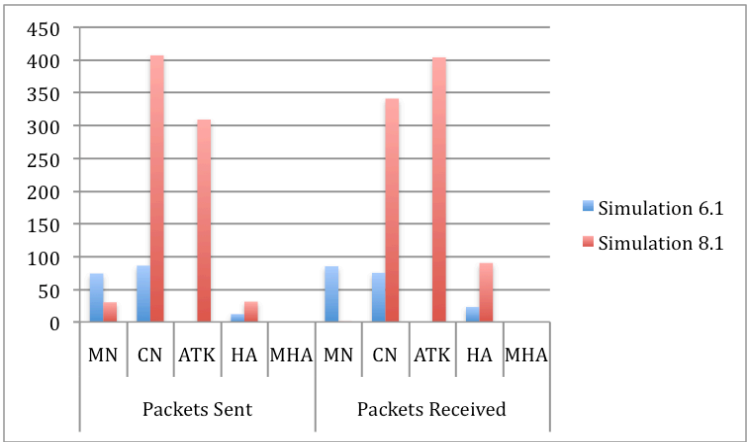

Figure 39. Comparison of RR Control Simulation 6 with RR Attack Simulation 8

Here we can see in Figure 39, that the Attacker is successful in obtaining both security tokens allowing it to create the binding key. Once it obtains the binding acknowledgement from the Correspondent it sends an update to the Home Agent telling it, it is the Mobile Node and its new address. Thereafter all traffic from the Home Agent is forwarded to the Attacker. The Mobile Node is unable to create the Binding key as the HoT token is forwarded by the Home Agent to the Attacker.

The next simulation tested the first of three proposed security solutions. The Distributed Authentication Protocol (DAP) is designed to work in combination with CGA and Return Routability but here we are testing its impact on the network without those components.

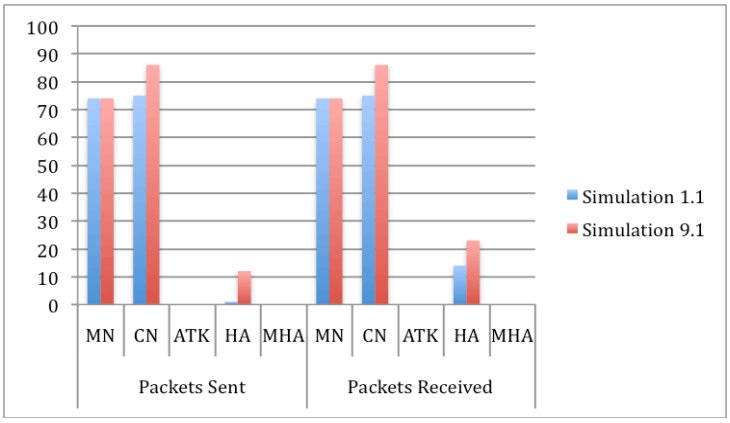

Figure 40. Comparison of Control Simulation 1 with DAP Control Simulation 9

Comparing DAP in a network without an attacker (simulation 9.1) to the control network (simulation1.1), the results, in Figure 40, show that there is minimum impact on the amount of data traffic in the network and only a slight increase in the amount of packets sent and received by the Correspondent and Home Agent.

DAP request authentication data from the Home Agent and the Mobile Node, but how does it handle when an attacker strikes?

Figure 41 shows that the communication with the Mobile Node is unaffected by the attack. This is because the Attacker is unable to authenticate itself with the Correspondent, as its data does not match that which is provided by the Home Agent.

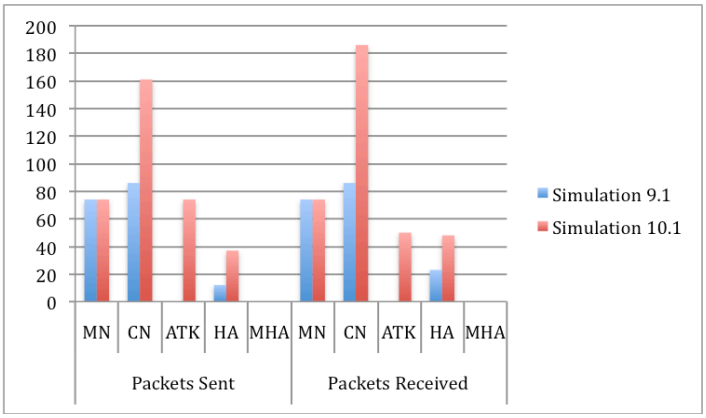

Figure 41. Comparison of DAP Control Simulation 9 with DAP Attack Simulation 10

The next simulation tested the second proposed security solution Dual Identity Return Routability (DIRR). This is a modification to the standard Return Routability simulated earlier in simulations 6, 7 and 8. First simulation to be performed tested DIRR in control conditions and is compared to standard network traffic and to a network with Return Routability.

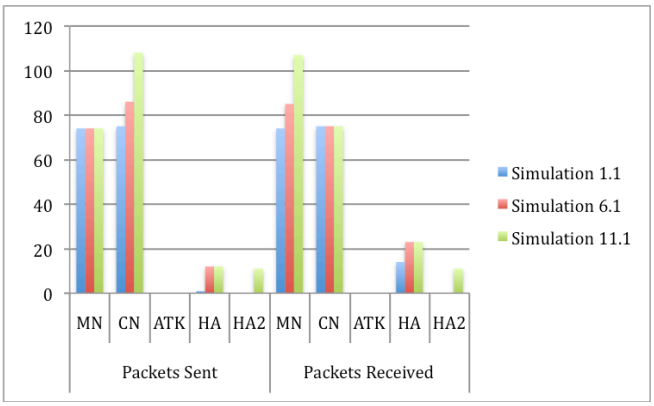

Figure 42. Comparison of Control Simulation 1 with RR Control Simulation 6 and DIRR Control Simulation 11

The graph in Figure 42 shows that in all three simulations the Mobile node sends the same amount of data in 60 seconds. It also shows that the correspondent receives the same amount of packets. However it can be seen that the correspondent sends and the Mobile receives more packets in Return Routability than in the control network and again in Dual Identity Return Routability even more data. This is true but not the whole story. Remember with this solution the second pair of tokens are sent over another network. This means that although more packets are sent and received to the Mobile and Corresponding nodes, from the point of view of the network there is the same amount of traffic. To test this, the simulation was run again which showed that 11 packets had been sent and received on the second networks' home agent. The results have been added to the HA2 column in the graph (for the purpose of this analysis) just to demonstrate that there is data on the second network comparable to that of the Home agent, which is being forwarded to the Mobile node. Multiply this by two and this takes into account both security tokens. This combined amount is what gives the Correspondent and Mobile Node an increase in the amount of data sent and received but without increasing the data traffic of any one network.

The next simulation tests DIRR against an attack. The attack is the same scenario, which the standard Return Routability faced in simulation 7.

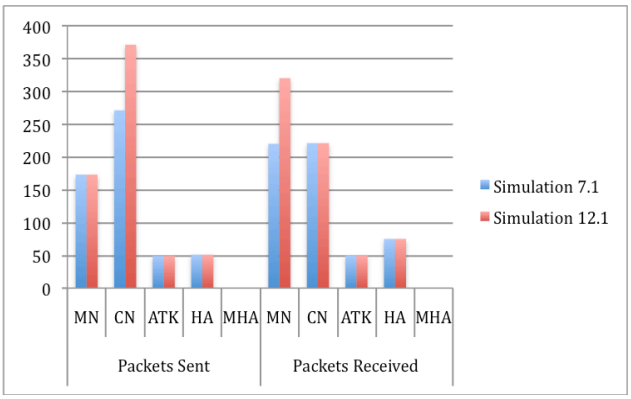

Figure 43. Comparison of RR Attack Simulation 7 with DIRR Attack Simulation 12

The graph in Figure 43 shows that the proposed solution (simulation 12.1) is just as effective as Return Routability (simulation 7.1) in securing the network in this scenario. However it also shows that there is an increase in network traffic and the amount of data sent and received by the Mobile and Correspondent.

Looking at the graph in Figure 44 we can see that the Mobile and Correspondent has nearly two to three times the amount of data packets sent and received.

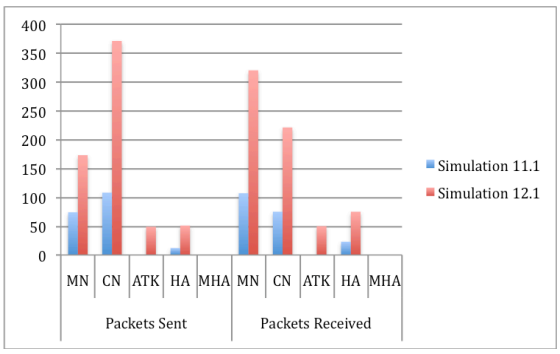

Figure 44. Comparison of DIRR Control Simulation 11 with DIRR Attack Simulation 12

Taking in to account that two security tokens are sent via the original network and two via an alternative network, there still seems to be a large amount of data being sent to the Mobile node from the correspondent. The reason for this is because when the Attacker request to connect with the Correspondent, the CN sends four tokens to what it believes is the Mobile node. It this scenario the Attack is only able to intercept one of the tokens, the other three being sent to the Mobile node, which is not expecting them so discards them.

This may seem as if four tokens is unnecessary as the solution has the same result as Return Routabilty. This is true for this scenario however let's test out an attack where the Home Agent has been spoofed.

In simulation 13 the Attacker spoofs the addresses of both the Mobile Node and the Home Agent.

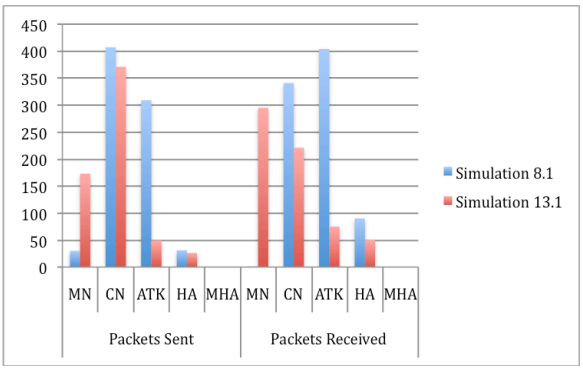

Figure 45. Comparison of RR Attack Simulation 8 with DIRR Attack Simulation 13

The graph in Figure 45 shows that Return Routability in simulation 8.1 was unable to protect against an attack of this nature however Dual Identity Return Routability can protect against this attack. The reason for this is that the Attacker would need to obtain all four tokens that are generated by the Correspondent however it can only obtain two of them. As the other two tokens are sent via an alternative network and via an alternative Home Agent the Attacker will have no way to access these packets.

This shows that in this attack scenario Dual Identity Return Routabilty is a more robust security solution than Return Routability.

In real world scenarios CGA and Return Routability are used together to provide grater protection.

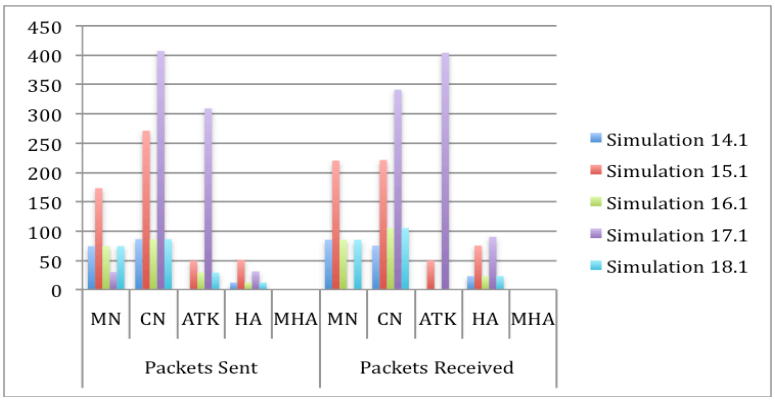

Figure 46. Comparison of Combined CGA and RR Simulations

In Figure 46 we can see simulation 14.1, which is the control network for the combined CGA and RR. The others are simulations of how the two solutions cope with a variety of combined attacks designed to bypass CGA and RR.

Simulation 15.1 shows that the attack failed. The Attacker did manage to bypass the CGA security by using its own CGA address but could only get one of the two tokens from Return Routability and the other went to the Mobile Node, which explains why it has a higher number of packets received.
Simulation 16.1 shows that this attack also failed as the CGA of the Attacker was denied authentication resulting in close to normal network performance.
Simulation 17.1 shows that the Attacker successfully managed beat both CGA and Return Routability by using it's own CGA address and spoofing the address of the Home Agent. This allowed it obtain both security tokens to generate the binding key and receive the Binding Acknowledgement which allowed it to update the Home Agent with it's current care of address, hijacking the communication away from the Mobile Node and preventing it from communicating with the Correspondent in a Denial of Service Attack.
The Attacker of simulation 18.1, just like in simulation 16.1 fails in its attack as it is unable to authenticate its CGA address to the Correspondent, which denies it access.

Taking these results as the bar in which to compare the proposed solutions, the next simulations in Figure 47 will add Distributed Authentication Protocol to CGA and RR to see if there are any improvement in securing against the same attacks.

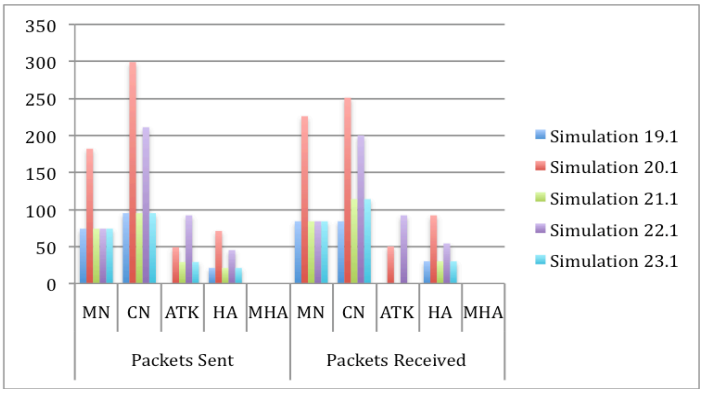

Figure 47. Comparison of Combined CGA, RR and DAP Simulations

Simulation 19.1 is the control without any attack.
The attack in simulation 20.1 manages to bypass CGA but is stopped by Return Rouability as it is unable to obtain both security tokens. This is the same outcome as in simulation 15.1 with CGA and RR.
Simulation 21.1 and 23.1 both show that the Attacker failed to get past CGA authentication, which is the same result as simulations 16.1 and 18.1.

However in simulation 22.1 the Attacker does manage to get past both CGA and Return Routability. However this attack is prevented by the Distributed Authentication Protocol because the Attackers authentication data does not match that provided by the Home Agent. Figure 48 shows that with DAP the mobile node can continue to communicate with the Correspondent and that data traffic generated by the Attacker it kept to a minimum.

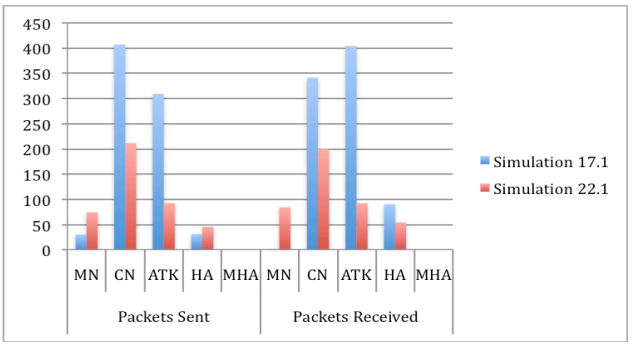

Figure 48. Comparison of Combined CGA & RR with CGA, RR & DAP

This is a clear advantage over the attack scenario seen in simulation 17.1 and shows that the addition of DAP to CGA and RR provides a last line of defence against certain attacks.

The following simulations, shown in Figure 49, will test Dual Identity Return Routability in combination with CGA and DAP.

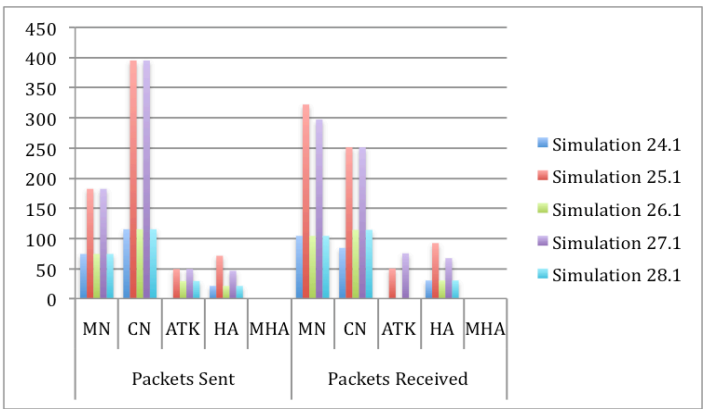

Figure 49. Comparison of Combined CGA, DIRR and DAP Simulations

Simulation 24.1 provides the results of the solution without any attack for comparison.
In simulation 25.1 the Attacker manages to bypass CGA but is only able to obtain one of the four security tokens from DIRR. This is the same result as simulation 20.1.
Simulations 26.1 and 28.1 show that the Attacker failed to authenticate its CGA address so was unable to attack the communication between the Mobile Node and Correspondent.
Simulation 27.1 shows that the attack was stopped by DIRR. The attacker bypassed CGA authentication and managed to obtain two of the security tokens however it needs all four tokens to create the binding key and those other tokens are transmitted on another network.
Comparing simulation 27.1 with 22.1 we can see that both these combined solutions are robust and secure.

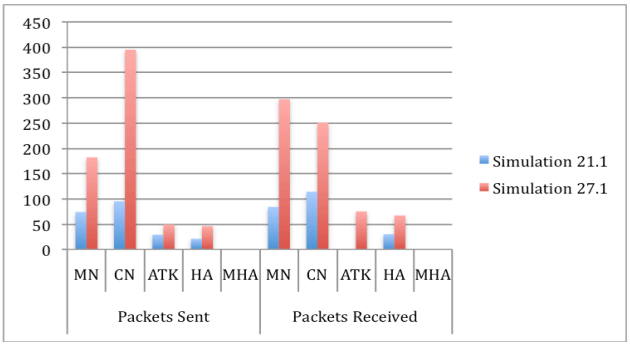

Figure 50. Comparison of Combined CGA, RR and DAP with CGA, DIRR and DAP

The graph in Figure 50 shows that the introduction of DIRR in simulation 27.1 does increase the network traffic considerably, however it should be noted that half the tokens are sent via an alternative network.

27.1 is a stronger solution as it managed to stop the Attacker with its second security component of DIRR while with simulation 21.1 the Attacker managed to get passed the second security component of RR and was stopped by the third component DAP. A possible solution to reducing the data traffic of simulation 27.1 would be to introduce DAP as the second component as it is less network intensive than DIRR. If the Attacker fails then the four security tokens would not be sent, as it would be the third component of the solution only taking place once the second had passed.

The final solution to be simulated was Mobile Home Agents. The purpose was to reduce the latency that packets incur during triangle routing and add an extra level of security to the Mobile Node.

The first test was to gauge a base line reading to compare to normal network data traffic and operations. Mobile Home Agents only work when the Mobile Node is away from the home network. The Home Agent sends a software copy of itself to a point of attachment close to the Mobile node, which continues to act as the Home Agent.

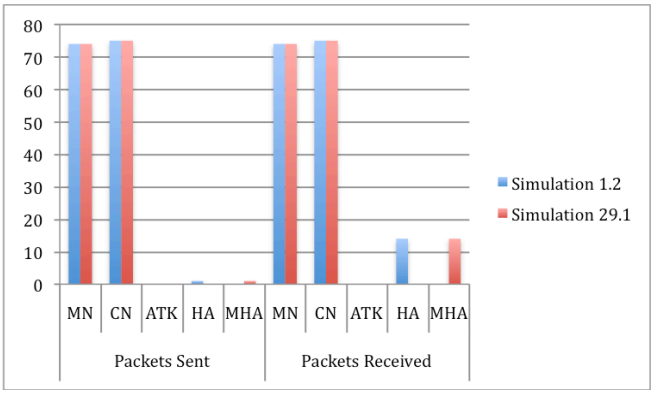

Figure 51. Comparison of Control Simulation 1 with MHA Control Simulation 29

The simulation in Figure 51 shows that the MHA has no adverse impact on normal network functioning and that the Mobile Home Agent successfully takes over the duties of the actual Home Agent when needed.

Even though there is no impact on the amount of data in the network this does not tell us the whole story. The following graph shows us the minimum and maximum hop count the data packets took to reach their destinations.

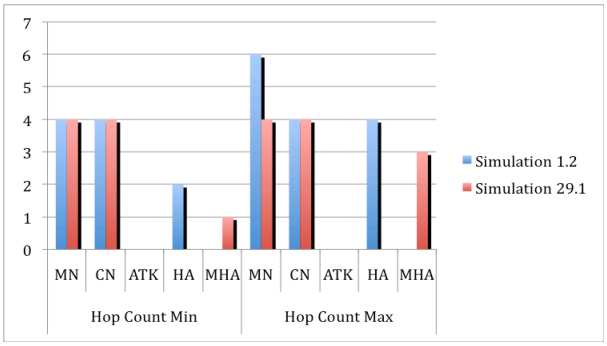

Figure 52. Comparison of Hop Count in Control Simulation 1 with MHA Control Simulation 29

Comparing the data between the HA and MHA for Min and Max hop count in Figure 52 we can see that the Mobile Home Agent has a lower hop count. There is also a much lower maximum hop count for the Mobile Node. This shows that over time the introduction of the Mobile Home Agent reduces communication latency between the Mobile Node and Correspondent.

The latency reduction is a welcome addition but are there any security benefits also?

Simulation 30.1 tests the Mobile Home Agent in a network with no security and Attacker performing an attack as was done in Simulation 2.

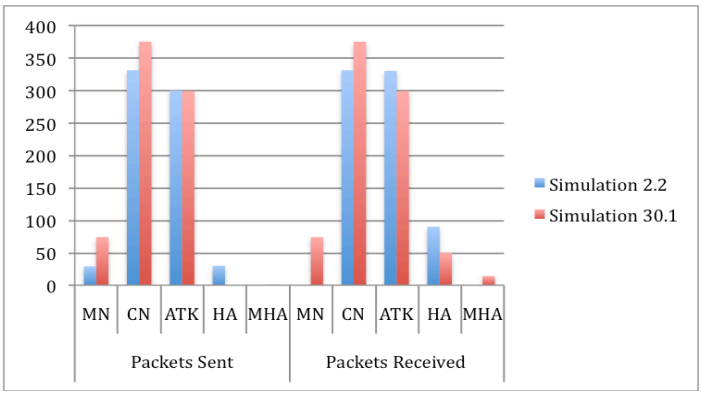

Figure 53. Comparison of Attack Simulation 2 with MHA Attack Simulation 30

The graph in Figure 53 shows that the use of the Mobile Home agent does not stop the attack from taking place. However it does prevent the denial of service attack from taking place as the Attacker is

unable to redirect packets to its self from the MHA. The packets received for MN is non-existent in simulation 2.2 but continues to receive packets in simulation 30.1 This can be considered as a partial success as the Mobile Node can continue uninterrupted.

Generally the addition of a Mobile Home Agent will have no impact on the successful security results that have been previously tested. But what about the simulations which showed the attacker mounting a successful attack?

In simulation 32.1 MHA was used in a network using CGA. In the same scenario in Simulation 4 the Attack successfully bypassed the CGA security and redirected all traffic to itself.

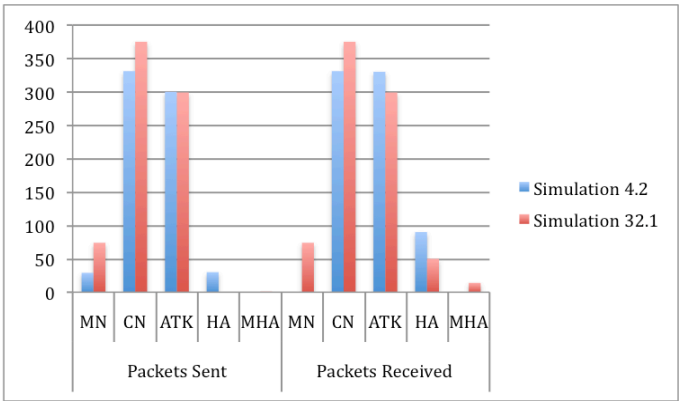

Figure 54. Comparison of CGA Attack Simulation 4 with MHA CGA Attack Simulation 32

However in simulation 32.1 in Figure 54 we can see that Attack is still successful but is unable to prevent the Mobile node from continuing to communicate with the Correspondent. Another partial success.

The next test in which the Attacker was successful was simulation 8 where it managed to obtain both tokens from Return Routabilty and hijack the communication.

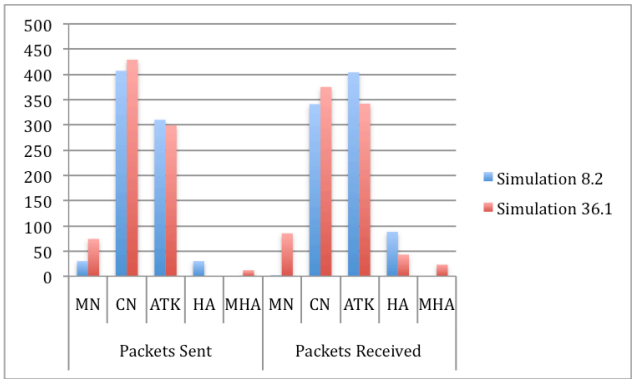

Figure 55. Comparison of RR Attack Simulation 8 with MHA RR Attack Simulation 36

Simulation 36.1 in Figure 55 shows that it too has partial success as it prevents the denial of service attack against the Mobile Node but the Attacker is still able to impersonate the Mobile Node.

The final partial success was seen in simulation 45.1.

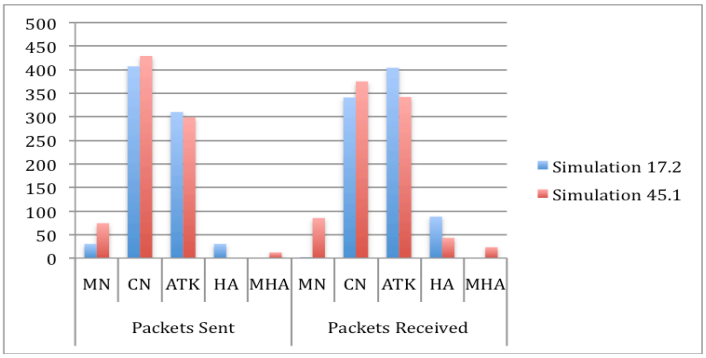

Figure 56. Comparison of Combined CGA & RR Attack Simulation 17 with MHA CGA & RR Attack Simulation 45

Figure 56 shows the combined CGA and RR of simulation 17 in which an attack scenario, redirected all data to the Attacker. This is compared to simulation 45.1 in which the introduction of the MHA has prevented the DOS attack however the attacker is still able to impersonate the Mobile Node.
As both nodes are communicating with the Correspondent, and the Attacker is impersonating the Mobile Node, the introduction of duplicate communication detection on the Correspondent could help to add an extra layer of security to these scenarios.

Finally how does the introduction of the Mobile Home Agent effect the full implementation of the proposed security solution which includes the combination of Cryptographically Generated Addresses, Dual Identity Return Routability and The Distributed Authentication Protocol?

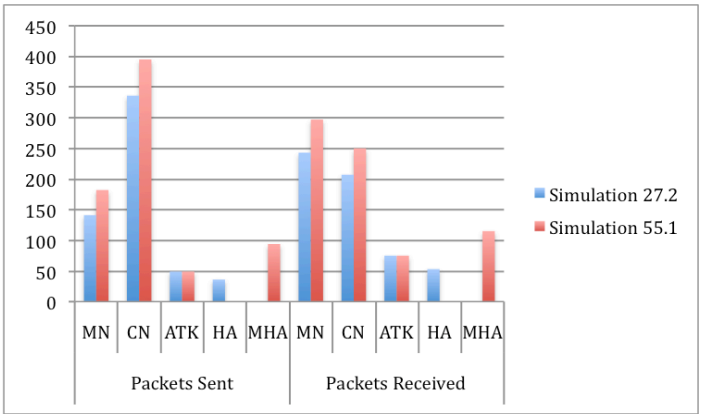

Figure 57. Comparison of CGA, DIRR and DAP Attack Simulation 27 with MHA CGA, DIRR and DAP Attack Simulation 55

In both simulation 27.2 and 55.1 the Attacker successfully bypasses CGA but is stopped by DIRR as it was only able to obtain two of the four tokens required to create the binding key. The outcomes of the two simulations are the same yet the introduction of the Mobile Home Agent gives us a different data traffic reading in the network.

Figure 57 shows that there is an increase in data packets being sent and received by the Mobile Node and Correspondent. This is because the latency of transmitting the packets between the nodes is reduced and so the nodes can reply faster and therefore are exchanging messages faster.

To support this explanation Figure 58 shows the minimum and maximum hop counts between the nodes of the two simulations.

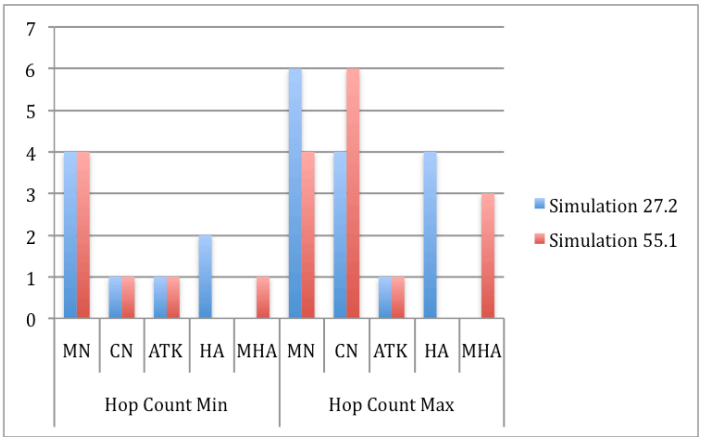

Figure 58. Hop Count Comparison of CGA, DIRR and DAP Attack Simulation 27 with MHA CGA, DIRR and DAP Attack Simulation 55

Here we can see that the Mobile Home Agent provides a shorter route to the Mobile Node by at least one hop. Over time this reduced latency improves data transmission and so the amount of packets exchange in the same period of time increases.

6.7 Conclusion

The simulations of proposed security solutions have been proven to be successful in preventing a variation of attacks that current security solutions are unable to defend against. Dual Identity Return Routability and the Distributed Authentication Protocol have both proven to resist attacks when combined together with CGA or used on their own. The addition of Mobile Home Agents has proven that it can prevent denial of service attacks and reduce the latency of communication between the Mobile Node and the correspondent.

Chapter 7 Analysis of proposed Solution

7.1 How the Proposed Protocol addresses the Security Vulnerabilities

No security protocol is 100% effective and in most cases the introduction of new security elements can solve the intended problem but can also increase the number of possible vulnerabilities in the system. The following chapter shows how the security protocol attempts to solve the security problems identified.

7.1.1 Non Authentication

Risks of Unauthenticated Binding Updates

Accepting an unauthenticated binding update leaves the node vulnerable to several attacks. The proposed security protocol uses three types of security which authenticates address ownership, location verification and device / user authentication. Therefore it is safe to assume that this problem is no longer a worry as no binding updates are accepted unless all three security checks are passed.

Risks of Unauthenticated Binding Acknowledgements

This is something that was overlooked but can be easily rectified. The final message of the protocol can be encrypted with the binding key or signed with the correspondent's private key.

Risks of Not Authenticating Home Agents

The protocol does introduce some methods, which do protect the home agent. Firstly home agent addresses are cryptographically generated protecting them from spoofing. Second, they must forward packets from the correspondent to the mobile node or return routability will fail and lastly the home agent must be able to provide a hash of the users authentication data or the authentication will fail.

Bogus on Link Prefix

An attacker can send a router advertisement message specifying that some prefix is on-link. The result is that it will never send a packet for that prefix to the router. The protocol does not address this problem however it could be resolved by only allowing messages of this type to be accepted after the user sending the message has been authenticated.

7.1.2 Denial Of Service

Denial of Service Attacks against Home Agents

An attacker may have two different home addresses and then send binding updates to bind them to each other as care of addresses, creating a routing loop. To protect against this attack home agents should only act on behalf of trustworthy mobile nodes, which it knows. This is solved in this case as mobile node would need a subscription with a service provider to use its home agent.

Risks of Not Verifying the Care-Of Address

The security protocol prevents DOS attacks by verifying that packets sent to a mobile's claimed care of address in fact reach the willing participant of the protocol, i.e the care of address of the mobile node, preventing redirection attacks.

Stateless Protocols

Stateful protocols can expose participants to denial of service attacks because the correspondent must store a separate state for each mobile it is in communication with and keep track of all the keys in use. The attacker can initiate the protocol many times causing the host to store large amounts of unnecessary protocol states. Unfortunately this may be a vulnerability, which has not been addressed. However additional modifications may reduce the risk to this vulnerability. Introducing a time to live for the completing of an authentication protocol can help prevent DOS occurring with endless authentication attempts. If an authentication attempt does not complete before the time to live ends then the authentication is abandoned.

Resource Exhaustion and Other Denial of Service Attacks

There are two types of denial of service attacks in mobile IP. Resource exhaustion where there is a limited amount of resources such as bandwidth or processing power and denial of service attacks such as forged binding updates. Forging binding updates is protected against due to CGA. Resource exhaustion however may be a problem and there are already a large number of network messages being passed for the protocol. The issue of processing power is side stepped with the introduction of distributed authentication.

Duplicate Address Detection DOS

When a host enters a network and uses stateless address auto-configuration to obtain an address, it is possible for an attacker to respond to every duplicate address detection attempt claiming that it owns

that address. This prevents the host from getting an address. This can be solved by CGA by asking the attacker to sign the message with his private key proving he owns the address.

Neighbour Discovery DOS Attack

An attacker can continuously send bogus addresses with a valid subnet prefix to the target network but with an invalid suffix. CGA resolves this problem as the attacker will not have the key associated with the address.

Consuming Authentication Resources

PKI are resource intensive and so this can be exploited by an attacker to drain the mobile of them by launching an attack that floods it with packets that need to be authenticated. The protocol does not use PKI for cryptography only for asserting ownership of addresses limiting this attack.

Reflection and Amplification

It is possible for an attacker to trick nodes in to sending the same number of packets, or more to a target. Due to the nature of this protocol and the high number of messages passed between nodes, this could be an effective attack if successful. However CGA and RR should prevent these attacks.

7.1.3 Redirection Threats

Threats from a Lack of Authentic Location Information

Redirection threats are removed with return routability as the nodes are checked to see if they can receive data.

Redirection (Bombing)

The protocol protects against false binding updates and so prevents denial of service attacks.

7.1.4 Masquerading / Spoofing

Malicious Last Hop Router

The attacker can masquerade as a last hop router by replying to a router solicitation or multicasting router advertisements. If it is selected then it will be able to redirect all traffic passing through it. Cryptographically Generated addresses prevent spoofing of legitimate addresses.

Neighbour Solicitation/Advertisement Spoofing

Cryptographically Generated addresses prevent spoofing of legitimate addresses

Spoofing Redirect Messages

Cryptographically Generated addresses prevent spoofing of legitimate addresses

Bogus Address Configuration Prefix

An attacker can advertise a false subnet prefix. The host executing auto-configuration will use the prefix to construct an address resulting in return packets never reaching the host. This is not addressed by the security protocol.

Parameter Spoofing

An attacker can duplicate a valid router advertisement but change the parameter values to disrupt traffic. Many of these attacks can be avoided with the use of authentication mechanisms, however even authenticated binding updates can be used to amplify a packet flooding attack.

Inducing Unnecessary Authentication

An attacker can exploit the binding update protocol by sending spoofed IP packets to the mobile that appear to come from different correspondents. Return routability and CGA prevent this attack. However a legitimate user could continually induce authentication unnecessarily. Future work should look in to a way of limiting the amount of authentication that can take place and under what circumstances.

7.2 Summary

The proposed security protocol is designed within the boundaries of the existing architecture and security technologies. The complimentary use for Cryptographically Generated Addresses and return routability limits the options for attackers. It does not however check authentication of the device or user. This has now been introduced and will enhance the security of existing systems. By not modifying any of the standards of mobile IPv6, the security solution should be compatible with any future implementation and at a low cost. The introduction of distributed authentication can have benefits under processor intensive situations but the drawback is that there is an increase in network messages. The option to choose between standard and distributed authentication is a useful choice however under which circumstances one or the other should be used.

8 Conclusions

8.1 Introduction

Wireless communication technologies have come a long way with improvements with every generational leap. As communications evolve so do the system architectures, models and paradigms. Improvements have been seen with jump from 2G to 3G in terms of security. Yet security issues persist and will continue to plague mobile communications in to the leap to 4G if not addressed. 4G will be based on the transmission of Internet packets only using architecture known as mobile IP. This will feature many advantages however security is still a fundamental issue to be resolved. One particular security issue involves the route optimisation technique, which deals with binding updates. This allows the corresponding node to by-pass the home agent router and communicate directly with the mobile node. The home agent has a static address while the mobile node's address changes every time it moves to a new location with a new point of attachment. The home agent keeps track of the mobile node's current address, so that if a correspondent does not know it, it may send the packets to the home address, which will forward the packets to the mobile node. By bypassing the home address with the binding update route optimisation, the speed of the delivery of packets will increase. There are a variety of security vulnerabilities with binding updates, which include the interception of data packets, which would allow an attacker to eavesdrop on its contents breaching the users confidentiality or to modify transmitted packets for the attackers own malicious purposes. Other possible vulnerabilities with mobile IP include address spoofing, redirection and denial of service attacks. For many of these attacks all the attacker needs to know is the IPv6 addresses of the mobile's home agent and the corresponding node.

Numerous security solutions have been proposed and each have their advantages and disadvantages. The two main types of security are encryption and authentication. Encryption protects the confidentiality of the data and comes in two flavours, symmetric and asymmetric. The former is useful for low powered devices and participants use the same key to encrypt and decrypt. The problem is how to distribute the key without it being intercepted. Asymmetric keys are split in to encryption and decryption keys. This is useful for the distribution of the keys and can help with authentication with the use of digital signatures. The drawback however is that processing consumption is 100 – 1000 times that of symmetric cryptography. This can be reduced somewhat with the implementation of elliptic curve cryptography which is a lightweight public key cryptographic solution.

Authentication allows users to verify that they are communicating with validated participants. Different systems exist, such as Kerberos that perform authentication by referring to a central authentication database to compare users credentials. However in the mobile IP architecture it is best to stay away from centralized authorities as they are a single entity and hence a single point of attack. A distributed authentication system is required which use techniques such as hashes, digital signatures, address based

keys and cryptographically generated addresses. Address based keys and certified addresses could be used for signing address resolution, duplicate address detection and redirection messages however, cryptographically generated addresses have the advantage that no trusted third parties are required
More elaborate systems such as RADIUS based on AAA Authentication, authorization and accounting, allow for a combination of security but must rely on an authentication server. Currently it is recommended that all mobile IP security be handled by IPSEC. However the cost in resources to utilize IPSEC is beyond what it realistically expected from a mobile device in effect reducing the users quality of service.

Security protocols have been specifically designed for the protection of binding updates such as, Bake/2 and CAM, from eavesdropping modification and DOS attacks. However they all make the same fundamental error. They give away the location of the home agent, the mobile node and the correspondent. This is the basis for many of the possible attacks to these nodes.

A solution must be developed that allows for the crucial location information to be transmitted and yet the nodes retain their location privacy. Systems have been developed that do this at some level such as the hierarchical mobile IP management model's use of the mobility anchor point (MAP). However location privacy is moot point with the introduction of cryptographically generated addresses. This allows users to assert their ownership over an address preventing spoofing. This combined with return routability may provide secure solution.
What is needed is a system that can fulfil the security needs of mobile IP's vulnerabilities by using a combination of the security technologies available, which operate without over taxing the computing resources available and package them into an easy to implement solution.

The proposed protocol utilizes a combination of established and innovative security solutions. It has been design to work within the existing infrastructure without modifying the standard architecture. The combined use of Cryptographically Generated Addresses and Return Routability provides address ownership and reachability validation. However the lack of authentication has been resolved with the introduction of the Distributed Authentication Protocol, which provides a low cost solution with benefits under processor intensive situations. The fact that the protocol has not modified the standards of Mobile IPv6 means that it will be compatible with future implementations with little to no modification necessary. The only drawback of the protocol are the increase in network messages however the user can choose if they require the distributed feature of the solution and choose not to use it under certain circumstances.

8.2 How were the key research questions addressed?

Can a security solution be developed that solves the security issues of binding updates in Mobile IPv6 using the existing infrastructure?

The proposed solution of the Distributed Authentication Protocol was created to address this question. The use of a Home Agent to store the hash of the Mobile Nodes identification data allows authentication to take place in a distributed manner without the need for a central authentication authority. This is secure, as the Home Agent would be provided by the telecom / Internet Service Providing company which the correspondent will be familiar with. The use of hashed data means the solution can operate with very low processing requirements.

Can an existing security solution be enhanced to improve its effectiveness and make it less vulnerable to certain types of attack?

Dual Identity Return Routability attempted to address this question by enhancing the existing solution Return Routability. The standard solution sends two tokens to the Mobile Node, one via the Home Agent and the other directly. The Mobile Node then combines them together to create the binding key. The simulation showed that this was vulnerable to attack if the Attacker managed to acquire both tokens. Dual Identity Return Routability made this type of attack much more difficult by giving the Mobile Node two identities, each with their own connection to separate networks and using four security token instead of two. Two of the tokens would work in the same way as the standard solution but the other two would be sent to the second identity, which would be on an other network and have it's own Home Agent. This makes it very difficult for an Attacker to mount an attack, as it would require all four tokens to be intercepted. Once the Mobile Node receives the tokens then the binding key is created and sent to the correspondent. The use of the second identity takes advantage of a device in which the user could have a separate private and business use.

Can an enhancement to the mobile IPv6 infrastructure be introduced to reduce the latency of triangle routing packets to the Home Agent while maintaining or enhancing binding update security and location privacy?

Mobile Home Agents was proposed as a solution to this question. Before route optimisation takes place communication between the Mobile Node and correspondent takes place via the Home Agent. This can begin to introduce communication latency the further away the Mobile Node travels from the Home Network as packets would be sent to the Mobile Node first before being tunnelled to the Mobile Node. The introduction of Mobile Home Agents resolves this issue by creating a software agent, which emulates the functions of a Home Agent and operates from the point of attachment where the Mobile Node is located. As the Mobile Node re-attaches it's self to a new point of attachment the Mobile Home Agent stops, migrates to it and then resumes its functions.

The advantage of this is that even though triangle routing is still taking place, the packets will travel via the Mobile Home Agent, which is in the same location as the Mobile Node, reducing the latency of communication, as the Mobile Home Agent will allow communication to occur at close to direct connection speed.

Security is increased as the Mobile Home Agent acts as a proxy and firewall for the Mobile Node to the network. It was observed in the simulations that the introduction of the Mobile Home Agent prevented Denial of Service Attacks in certain scenarios. This was due to the Attacker trying to redirect packets from the Home Agent as it was unaware of the Mobile Home Agent. This allowed the Mobile node to continue to communicate with the Correspondent even when the Attacker mounted a successful false binding update attack.

As the Mobile Home Agent is in the same location as the mobile node, it is feasible to give its address to nodes on the public network as if it was the Mobile Node. Then any traffic arriving will be tunnelled to the Mobile Node. This adds another layer of protection as it hides the true IP address of the Mobile Node, which protects it from a direct attack.

8.3 Main Contributions

The thesis provides three unique contributions:

A. **The Distributed Authentication Protocol.** This provides a de-centralised authentication solution within the existing Mobile IPv6 infrastructure, which can be used on its own or in combination with other security techniques.

B. **Dual Identity Return Routability.** This is an enhancement to Return Routability, which provides location authentication by using a second identity on an alternative network to transport half the security tokens needed to create a binding key.

C. **Mobile Home Agents.** This is a software agent, which emulates the functions of the Home Agent and operates from the point of attachment of which the Mobile Node is located. When the Mobile Node moves to another point of attachment, so does the Mobile Home Agent. This provides a layer of security to the Mobile Node, as the Mobile Home Agent can't be as easily attacked, due to its mobility and dynamic nature, unlike that of a static Home Agent. This allows for binding updates to be less susceptible to types of Denial of Service attack. It also provides a reduction in communication latency for packets destined for the Home Agent, as those packets are no longer routed to the home network first.

8.4 Elaboration on the main contributions

A. The Distributed Authentication Protocol.

There are three main aspects to the security protocol:

1. Cryptographically Generated Addresses
2. Return Routability
3. Authentication verification

The first two technologies are well-established techniques. Cryptographically Generated Addresses provide a reasonable assurance that the address of the user is indeed owned by them and not spoofed. Return Routability provides location authentication proving the communicating device is at the IP address claimed and again combats spoofing.
The third aspect of the security protocol provides solid device authentication and can be expanded to include user authentication in case of device or identity theft.

Adding security features means that there will be an increase in processing power needed by devices. To aid with this burden the protocol proposes using a distributed authentication architecture. The correspondent Node requests authentication data from the Home Agent and the Mobile Node. The Home agent stores the data as a hash which is which is unreadable by any attacker who would intercept it when transmitted. The Mobile Node sends the plane text data, which of course could be intercepted. To protect the plain text data on its way to the correspondent, the Mobile Node can encrypt it with the binding key created from the Return Routability stage.

Both pieces of authentication data are sent to the correspondent where the encrypted data from the Mobile is deciphered and then the data is hashed. The two strings are then compared and if they match the authentication process passes.

The Distributed Authentication Protocol provides a decentralised authentication system when there is no central authority. Each Mobile Nodes' security data is stored with it's own Home Agent which is maintained by the Internet Service Provider which manages it. This provides a safe and secure authentication infrastructure without any one single point of attack. If a Home Agent is attacked it will not effect anyone else using the system as each Mobile Node has it's own Home Agent.

The simulations show that the Distributed Authentication Protocol was successful in protecting the Mobile Node from false binding update attacks when Cryptographically Generated Address and Return Routabilty both failed. It is also relatively low latency and not processor intensive, as it required two messages, two replies, a decryption, a hash and a string comparison. The most

processor intensive aspect would be the encryption/decryption process but as the decryption is done on the Correspondent, in most cases it will be a large server computer with enough resources to handle the processing requirements. Only the Mobile Node when encrypting the authentication data with the binding key could experience any delay in processing as mobile devices could have limited resources. However, the reality of Modern Mobile Devices is that processing power and memory availability is the same if not more of what Desktop Computers where a few years ago.

B. Dual Identity Return Routability.

This is an enhancement to Return Routability, which provides location authentication by using a second identity on an alternative network to transport half the security tokens needed to create a binding key.
This solution demonstrates that dual identity phones can be used to improve security within 4G networks.

Dual Identity Return Routability is part of a larger security solution, but could be used as a stand-alone solution. Taking Return Routability as a template, it was modified by adding the feature of transmitting another two security tokens to the Mobile Node via an alternative network.
The Mobile Node would have two identities both connected to the network. This would be for example one phone number connected to O2 and the other connected to Orange. The four packets would arrive at the Mobile Node via their different paths, which would then combine the together to create the binding key.

The only drawbacks of this method is that it would require a modification to the network infrastructure and the increase in network traffic. The Mobile Node would have to be designed with the ability to have two identities and the software would have to be developed to control and manage both of them. Phones have been modified with the use of dual sim cards and have the ability to use both at the same time. They are called Active Dual Sim phones and both sims are active and the mobile device has two transceivers allowing the user to receive calls from either number at any time. As the technology is available and relatively cheap to produce, there is would be a very low price point in introducing this technology into 4G mobile networks and mobile telecom companies would benefit as it would over night double the potential amount of customers.

C. Mobile Home Agents.

Mobile IPv6 provides two methods of communication between the mobile and correspondent node. The first is triangle routing, which is when all communication to the mobile node is via the home agent. This is necessary as the home agents' IP address is static and is the first point of contact for any communication to the mobile node. The disadvantage however is that the further the mobile

node travels from the home agent the further data packets will have to travel to reach their destination.

The second method involves the use of a route optimization technique, which allows direct communication between the mobile and correspondent node. This is achieved with the use of binding updates. The disadvantage to this method is that the location of the mobile node is revealed to any correspondent in communication with it, which could be a potential security risk.
The thesis introduces an alternative method which provides the best of both worlds without the disadvantages.

The concept involves the introduction of mobile agent technology into mobile IPv6 networks.
The way they would be used is as an intermediary between the mobile node and the correspondent effectively becoming triangle routing. However the mobile agent would reside on the IPv6 node, which the mobile node is using as its point of attachment. The mobile agent is a piece of software responsible for routing messages from other nodes to the mobile node and at the same time provide location privacy by acting as a proxy and masking the true IP address of the mobile node.

As the mobile agent resides on the mobile nodes point of attachment there is negligible latency in comparison to triangle routing via the home agent. As the mobile agent will effectively resume most of the roles of the home agent we can call it a mobile home agent. But why is it mobile?

As it resides on the mobile nodes point of attachment, if the mobile node travels to a new location it will connect to a new point of attachment which will then be responsible for the mobile node as all communications are handed over to it. However the mobile home agent would not lose communication with the Mobile Node as the software is autonomous and capable of duplicating itself to the new point of attachment and resuming its role in the network.

Every time the mobile node moves to a new point of attachment the mobile home agent will follow providing constant location privacy with the advantages of low latency communication.
The simulations proved that there was a reduction in communication latency with the use of Mobile Agents and also showed that denial of service attacks were no longer effective against the Mobile Node as the Home Agent was no longer static and not easy to locate to unauthorized nodes. This fulfilled the requirements for location privacy, reduction in communication latency and denial of service security. However, the Mobile Home Agent does not have any direct effect on securing against false binding updates or impersonation attacks as these can still take place but with a further addition to the security protocols the Correspondent should be able to detect multiple simultaneous communication streams from apparently the same user but from different addresses and take appropriate action.

8.4.1 Identification of crucial gaps in knowledge

The research carried out identified that there are several gaps in knowledge when it comes to the security of binding updates in Mobile IPv6.
The first gap identified was the vulnerability of the binding update message within the network. The message is crucial for the route optimisation to take place but it is also a prime target for redirection, impersonation, man in the middle and denial of service attacks.

The security solutions that exist were analysed and it was determined that there were holes that attackers could take advantage of. This was proven in the simulation results.
With the introduction of the proposed security techniques, these holes no longer exist, however other gaps in knowledge have arisen.

1. What are the long-term security issues facing the distributed authentication protocol?
2. Are there any technical issues in trying to get the Dual Identity Return Routability to work across two networks simultaneously?
3. How robust will the programming be of a Mobile Home Agent to operate autonomously away from the home network and would they be able to cope with attackers using malicious Mobile Agents?

These are all questions, which could be answered, in future studies.

8.5 Future improvements to solutions from which the study can benefit

1. **The Distributed Authentication Protocol.**
   Improvements could take place in this protocol by possible combining this security step with other steps in the solution, therefore reducing the latency before a binding update can take place. However as it would be more secure to encrypt the Mobile Nodes authentication data with the binding key it would be prudent for this step to take place after Return Routability.
   Other improvements include the securing of the Home Agent as a repository for the Mobile Nodes authentication data and to make sure redirection attacks would not prevent the Home Agent from sending authentication packets back to the Correspondent.
2. **Dual Identity Return Routability.**
   For this solution to work the mobile device must have two identities either via two sim cards or a single sim card with both identities on it. Information needs to be processed from both networks for the solution to work, which would require the devices' operating system to be upgraded to handle this new feature and applications developed to take advantage of the possibilities this provides. The only draw back of the solution is the increase of data sent and

received, however as this extra data is sent via an alternative network, the load on the original network is the same.

3. **Mobile Home Agents**

   Mobile Home Agents would have to be developed with a very high level of artificial intelligence to handle autonomous operation in a potentially hostile foreign network. They must be able to reproduce the same service as they Home Agents while at the same time managing the tunnelling of data to the Mobile Node and it's own migration to the current point of attachment.

8.6 How can proposed solutions be applied in the real-world?

The solutions provided in this thesis will be of most benefit to two areas of mobile networking.
1, The next generation of mobile networks using 4G technology and 2, The securing of financial transfers in online mobile commerce.

8.6.1 Integration with 4G mobile devices

Mobile IPv6 networks have yet to be implemented in the public domain. However when they are the solutions should be relatively straightforward to implement.
The Distributed Authentication Protocol does not introduce any new hardware elements into the network. For the solution to be implemented software modifications would need to be integrated into the normal operations of the nodes. The Correspondent will need to ask for the authentication packets and Mobile Node and Home Agent would have to respond with their authentication packets.

The Implementation of Dual Identity Return Routability is more difficult as it requires new hardware components in the handset, however these can easily be introduced as a standard feature of 4G network devices which could be a popular feature helping in the uptake of the technology. With the feature as a standard then the security solution of using both networks simultaneously for location authentication will be a matter of applications deigned to receive and process data from both networks.

Mobile Home Agents could be developed relatively easily as they are just portable applications however the issue of their operation would be in the authentication of the software agents in a foreign network. The nodes in the foreign network would need to be modified to accept the use of a Mobile Agent and will need to give it access to it's resources.

8.6.2 Application to Mobile Commerce

The Internet has become a very popular method of purchasing goods and services and very competitive prices. As the Internet becomes more and more mobile, access to it becomes more ubiquitous in our

everyday lives. This means that the population understand the benefits and convenience of purchasing goods online and will do so more and more from mobile devices. However with the introduction of Mobile IPv6 networks the trust of using mobile commerce should not only be maintained, but enhanced. A robust secure networking environment is essential for the success of the technology.

The security techniques proposed in this thesis attempts to provide the world of mobile commerce with the ability to conduct purchasing of digital and physical goods in a safe and secure environment.
Cryptographically Generated Addresses provide address evidence of address ownership. Return Routability provides location authentication. Distributed Authentication Protocol provides user authentication. Dual Identity Return Routability enhances location authentication by protecting the token via another network and Mobile Home Agents reduce latency and prevent denial of service attacks. Together they provide the framework for a secure mobile e-commerce environment.

8.7 Limitations of the Research

There are numerous security attacks that can be mounted on any nodes in a Mobile IPv6 network. However the scope of this research will be specifically aiming to secure the binding update massage to ensure the robustness of the route optimisation technique.
The first proposed solution aimed to create a security framework, which did not need to introduce any new hardware or software elements in to the network.
The subsequent solutions did not have these restrictions and so introduced new hardware elements in the likes of dual sim technology and mobile software agents.

The limitations from a testing perspective is that Mobile IPv6 networks do not yet physically exist and so all the test had to be done in software simulations and programmed to behave as closely to the way the theory suggests they do. The simulations run could not be the full implementations of the proposed security solutions as certain elements of their operation were irrelevant for the purposes of security testing but are close enough to get valid data results.

An example of this is with Mobile Home Agents. The Agent is fully operational and can carry out the functions of the Home Agent however is not programmed with the migration ability to move between points of attachment as this was not the purpose of the simulation tests. The simulation assumes that the migration has successfully taken place and then simulations starts with the exchanges of the first messages. This keeps it consistent and comparable to the other simulations run.

8.8 What are the future works that can be pursued based on this study?

The research introduces three new unique security solutions to the Mobile IPv6 environment. The first is the Distributed Authentication Protocol, which introduces much-needed authentication in a distributed form. The second is Dual Identity Return Routability, which enhances reachability verification and third Mobile Home Agents, which provide a secure and optimised method of communication to and from the Mobile Node. They have been proven to improve the security and reliability of communication between the Mobile and Correspondent Nodes however some aspects of the protocols could be improved:

8.8.1 Future work

It must be noted that the home agent may be responsible for managing several hundred addresses. Therefore there is a limit to how much it will be able to process at one time. It would have to be calculated what would be the upper limit to the amount of address the home agent can be responsible for.

It is suggested in [1] that CGA can be improved by including the routing prefix of the network into the hash function:

Interface ID = $HASH_{64}$(Public Key | Routing Prefix)

This forces the attacker to perform the search separately for each prefix. Generating new public keys and regularly changing addresses increases the difficulty of brute force attacks. It is suggested in [18] that even more variables may be added in to hash to increase security such as link layer address as seen in Figure 59.

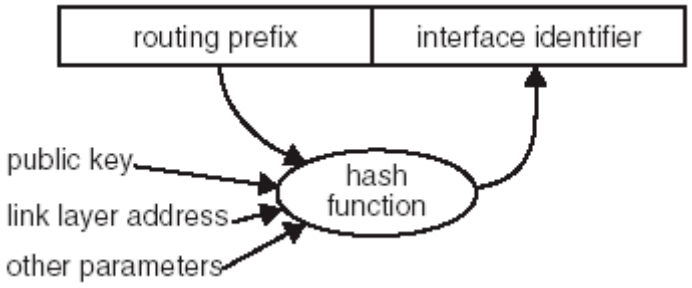

Figure 59. CGA hash function [18]

Optimisations to the transmission of packets would allow for reduced network traffic. Combining some of the messages together or finding a way of getting packets to there final destination without having to

pass through an intermediate node would be a vast improvement. A solution to this has been proposed with Mobile Home Agents.

8.8.2 Future Developments for Dual Identity Return Routability

Further work can be done in the development of Dual Identity Return Routability, which would allow for a wider adoption of devices with multiple network connections. This can be useful outside of security to allow a personal and business number to exist on the same device, which would allow the user to disable one of them if they chose to such as turning of the business identity when the user is at home. The other advantage it that in some areas cell tower coverage is limited or a telecom company may have a stronger signal. The use of two networks would allow the user to select whichever network provided a stronger signal.

8.8.3 Further work for Mobile Home Agents

Further research could be done in Mobile Home Agents in development in the migration protocols to aid in the operation of moving from point of attachment to point of attachment. This may open up possibilities in the research areas of artificial intelligence as the Mobile Home Agent will have to deal with events on its own in a foreign network.
There is a possibility that attacker could see the introduction of the Mobile Home Agent as new potential opportunity for attacks within a network if perhaps they introduced their own spoofed Mobile Home Agent. This could be prevented however with an authentication mechanism which verified the authenticity of the Mobile Home Agent perhaps by confirming with the Home Agent and Mobile Node that they are expecting the Mobile Home Agent to be operating. This could be another area of research that could be potentially looked in to.

The proposed solution of Mobile Home Agents fulfilled the requirements for location privacy, reduction in communication latency and denial of service security. However, the Mobile Home Agent does not have any direct effect on securing against false binding updates or impersonation attacks as these can still take place but with a further addition to the security protocols the Correspondent should be able to detect multiple simultaneous communication streams from apparently the same user but from different addresses and take appropriate action.
This could be a potential research area as detection of impersonation attacks could be useful in numerous fields of study and industry.

8.8.4 4G GPS Point of Attachment Location Authentication

A possible addition to the security solution would be the inclusion of GPS technology to help with location authentication. The basic premise is that the mobile node would send it’s GPS co-ordinates to

the corresponding node which would then in turn request the GPS co-ordinates from the mobile nodes current point of attachment. The co-ordinates would then be compared and if the mobile nodes co-coordinates fall within the proximity of the point of attachment then this proves that the mobile node is not spoofing its location or using proxies to access the network.
This can be combined with the other features of the distributed authentication protocol to create and even more robust security solution.

Initially this possible future security improvement may seem expensive to implement as each device and point of attachment would need a GPS system. However in reality there would be negligible cost in the implementation of this solution as more and more modern mobile devices possess GPS as a feature for use with applications such as maps or local services.

There would also be no hardware cost for the points of attachment either. This is because the vast majority of points of attachment are radio receiver/transmitters or wifi routers, which are not portable, and a physically attached to a single location. So a GPS devise would not be efficient in this case. All that would be needed is a one time calculation of the GPS co-ordinate at the position of the point of attachment. This can then be input and stored in to the routers memory for future use when required.

8.9 Concluding remarks

And so the research comes full circle. Beginning with the investigation of the next generation networks of Mobile IPv6, it was discovered that there was a vulnerability in the route optimisation procedure which could lead to a variety of attacks. The types of attacks were then investigated which demonstrated how devastating the consequences of not securing the binding update could be. The research then moved on to find out the current security techniques, which exist or could be used to resolve the situation however the research revealed that the solutions available did not provide sufficient protection to a wide range of scenarios. The advantages and disadvantages of the solutions were analysed and the best parts were taken and enhanced to provide a secure and robust solution.

The first solution proposed an addition to the Cryptographically Generated Addresses and Return Routability solution by adding user authentication in the form of the Distributed Authentication Protocol. This provided a decentralised, low latency, low resource requirement authentication system which when tested in the simulation proved that it was capable of withstanding attacks that could bypass both CGA and Return Routability. It was also designed to be used in a network with the minimum of modifications required.

The second proposed solution enhanced the Return Routability solution by introducing a second identity. This would be using another network connection and the theory was that sending security tokens to the Mobile Node via the two identities will secure the transport of the data packets as it would

be unlikely that all of them could be intercepted and that the combined use of two identities will make identity theft more difficulty to achieve as both identities are linked together. The simulation results show that Dual Identity Return Routabilty to be a more robust and secure solution to the standard Return Routability.

The introduction of Mobile Home Agents provided the ability for low latency communication between the communication nodes and as the simulation proved, provide protection against denial of service attacks.

The solutions proposed attempted to implement solutions, which could be utilised on any low powered mobile device and assumed the devices would have limited resources.
However, the reality of Modern Mobile Devices is that processing power and memory availability is the same if not more of what Desktop Computer where a few years ago.

When this research began, Mobile Devices were very limited in resources and so consideration would have to be made for how effective the device could be while still being within the limits of the devices capabilities. However with the introduction of smart phones and portable tablet computing this is no longer an issue which opens up doors in creating even more powerful security but as is the case with cat and mouse chases, it would only be a matter of time before the hackers find an alternative method to attack the networks and devices and the whole security life cycle would begin again.

The next generation of mobile devices will need the next generation of security and hopefully this research has provided some helpful contribution towards this goal.

Looking at the emerging standards of the next generation of mobile networks, such as Long Term Evolution (LTE), we can see that real world applications of the proposed security protocol are valid as LTE is based on Dual Stack Mobile IPv6 (DS-MIPv6) which is a modified version of Mobile IPv6, which the protocol was designed for. The only difference being that DS-MIPv6 provides support for both IPv4 and IPv6 devices to tunnel packets over the same network. The proposed security can still be applied to this architecture with only minor modifications.

References

[1] Tuomas Aura, Michael Roe, and Jari Arkko. *Security of internet location management.* In Proc. 18th Annual Computer Security Applications Conference, pages 78-87, Las Vegas, NV USA, December 2002. IEEE Press.

[2] C. Perkins, Mobile IP Design Principles and Practices: Addison Welsey, 1998.

[3] Tuomas Aura. *Mobile IPv6 Security.* In Proc. Security Protocols, 10th International Workshop, Cambridge, UK, April 2002. Springer 2003.

[4] A.S Tanenbaum and M.V Steen, Distributed systems –Principle and paradigms, prentice hall, new jersey, 2002.

[5] Tuomas Aura, Pekka Nikander and Gonzalo Camarillo. *Effects of Mobility and Multihoming on Transport-Protocol Security.* In Proc. 2004 IEEE Symosium on Security and Privacy (SSP'04), Berkeley, CA USA, May 2004. IEEE Computer Society.

[6] Tuomas Aura, Pekka Nikander, Jussipekka Leiwo. *DOS-resistant authentication with client puzzles*. Proc. Security Protocols Workshop 2000, Lecture Notes in Computer Science, volume 2133, pages 170-181, Cambridge, UK, April 2000, Springer 2001.

[7] Integrity Sciences, Elliptic Curve Cryptography**,** http://world.std.com/~dpj/elliptic.html, 1997, last accessed august 2004.

[8] J. Kohl and C. Neuman, The Kerberos Network Authentication Service, RFC 1510, http://www.faqs.org/rfcs/rfc1510.html, September 1993

[9] J. Arkko, P. Nikander, and G. Montenegro. Selection of MIPv6 Security Level Using a Hashed Address Internet Draft draft-arkko-mIPv6-select-hash-00.txt. Work In Progress, IETF, June 2002.

[10] James Kempf, Craig Gentry, "Securing IPv6 Neighbor Discovery Using Address Based Keys (ABKs)." (IETF, May 6, 2003)

[11] Tuomas Aura. *Cryptographically Generated Addresses (CGA).* In Proc. 6th Information Security Conference (ISC'03), volume 2851 of LNCS, pages 29-43, Bristol, UK, October 2003. Springer.

[12] J. Arkko, V. Devarapalli, F. Dupont. Using IPSec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents. RFC 3776, IETF, June 2004.

[13] C. Rigney et al, Remote Authentication Dial In User Service (RADIUS, http://www.faqs.org/rfcs/rfc2865.html, RFC 2865, June 2000.

[14] J. Arkko, P. Calhoun, E. Guttman, D. Nelson, and B. Wolff. AAA Solutions. Internet Draft draft-ietf-aaa-solutions-01.txt. Work In Progress, IETF, November, 2000.

[15] M. Roe, G. O'Shea, T. Aura, J. Arkko. Authentication of Mobile IPv6 Binding Updates and Acknowledgments, Internet Draft draft-roe-mobileip-updateauth-02.txt. Work In Progress, IETF, February 2002.

[16] Greg O'Shea and Michael Roe, Child-proof authentication for MIPv6 (CAM), ACM Computer Communications Review, 31(2), April 2001.

[17] Security Threats and Requirements white paper, 3G TS 21.133 version 3.1.0, 3rd Generation Partnership Project, http://www.3gpp.org, 1999.

[18] Jari Arkko, Tuomas Aura, James Kempf, Vesa-Matti Mäntylä, Pekka Nikander, and Michael Roe. *Securing IPv6 neighbor discovery and router discovery*. In Proc. 2002 ACM Workshop on Wireless Security (WiSe), pages 77-86, Atlanta, GA USA, September 2002. ACM Press.

[19] Jean Tourrilhes, Problems I've found with Mobile IP, http://www.hpl.hp.com/personal/Jean_Tourrilhes/MobileIP/known.html, April 1994, last accessed August 2004.

[20] Certicom, The basics of ECC, http://www.certicom.com/index.php?action=res,ecc_faq, August 2004.

[21] A. Okazaki et al, Securing mIPv6 binding updates with address based keys, IETF Draft, draft-okazaki-mobileip-abk-00.txt, June 2002

[22] R. M. Needham and M. D. Schroeder, ``Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM*, Vol. 21, No. 12, December 1978, pp. 993-999.

[23] Mike Just, Needham-Schroeder Protocols**,** http://www.win.tue.nl/~henkvt/Needham.doc, last accessed 2004.

[24] Brian Tung, The Moron's Guide to Kerberos, http://www.isi.edu/gost/brian/security/kerberos.html, December 1996, Last accessed August 2004.

[25] P. Eronen, J. Arkko. Authentication components: Engineering experiences and guidelines. Submitted for publication, February 2004.

[26] C.Metz, AAA protocols: Authentication, authorization and accounting for the internet. Cisco systems, IEEE Internet computing, December 1999.

[27] Webopedia, IPSEC, http://www.webopedia.com/TERM/I/IPSec.html, last accessed 2004

[28] S. Kent and R. Atkinson, Security Architecture for the Internet Protocol, RFC 2401, November 1998.

[29] D. Harkins and D. Carrel, The Internet Key Exchange (IKE), RFC 2409, November 1998.

[30] Jari Arkko, and Pekka Nikander, "Limitations of IPSec Policy Mechanisms," to appear in *Security Protocols, Eleventh International Workshop*, Cambridge, UK, April 2-4, 2003. 003.

[31] Pekka Nikander, "Denial-of-Service, Address Ownership, and Early Authentication in the IPv6 World," in Security Protocols, 9th International Workshop*,* Cambridge, UK, April 25-27 2001, LNCS 2467, pp. 12-26, Springer 2002.

[32] D. Johnson et al, Mobility Support in IPv6, RFC 3775, http://www.faqs.org/rfcs/rfc3775.html, June 2004

[33] Pekka Nikander, Tuomas Aura, Jari Arkko, and Gabriel Montenegro, "Mobile IP version 6 (MIPv6) Route Optimization Security Design -- Extended abstract," in *Proceedings of IEEE Semiannual Vehicular Technology Conference, VTC2003 Fall, IP Mobility Track,* Orlando, Florida, October 6-9, 2003

[34] C. Vogt, J. Arkko, R. Bless, M. Doll, T. Kuefner. Credit-Based Authorization for Mobile IPv6 Early Binding Updates draft-vogt-mIPv6-credit-based-authorization-00.txt. Work In Progress, IETF, May 2004.

[35] W. Haddad et al, Applying Cryptographically Generated Addresses to Optimize MIPv6, http://www.ietf.org/internet-drafts/draft-haddad-mip6-cga-omipv6-03.txt, October 2004

[36] Tuomas Aura, Carl Ellison. *Privacy and Accountability in Certificate Systems*. Research Report A61, Laboratory for Theoretical Computer Science, Helsinki University of Technology, Espoo, Finland, April 2000.

[37] T. Aura, Cryptographically Generated Addresses (CGA), Request for Comments: 3972, http://tools.ietf.org/html/rfc3972, March 2005.

[38] J. Arkko et al, Enhanced Route Optimization for Mobile IPv6, Request for Comments: 4866, http://tools.ietf.org/html/rfc4866, 2007.

[39] E. Nordmark et al, Shim6: Level 3 Multihoming Shim Protocol for IPv6, Request for Comments: 5533, http://tools.ietf.org/html/rfc5533, 2009.

[40] J. Laganier, Authorizing Mobile IPv6 Binding Update with Cryptographically Generated Addresses, draft-laganier-mext-cga-01, Internet-Draft, October 26, 2010.

[41] W. Haddad et al, BUB: Binding Update Backhauling, www.ietf.org/internet-drafts/ draft-haddad-mipv6-bub-02.txt, February 2004

[42] Jukka Ylitalo and Pekka Nikander, "BLIND: A Complete Identity Protection Framework for End-points", to appear in *Security Protocols, Twelfth International Workshop,* Cambridge, 24-28 April, 2004.

[43] Qi He et al, The quest for personal control over mobile location privacy, IEEE communications magazine, Vol. 4, No.2, May, 2004.

[44] A. Escudero, Location Privacy in IPv6: 'Tracking binding updates'. Tutorial at Interactive Distributed Multimedia Systems (IDMS2001). Lancaster, UK. September 2001.

[45] Baldi et al, "Exploiting Code Mobility in Decentralised and Flexible Network Management", in Mobile Agents, Lecture Notes in Computer Science 1219, Rothermel, K., Popescu-Zeletin, R., Springer-Verlag, Berlin. 1997

[46] Biehl, I et al. " Ensuring the Integrity of Agent-Based Computations by Short Proofs" in Mobile Agents, Lecture Notes in Computer Science 1477, Rothermel, K. and Hohl, F., Springer-Verlag, Berlin, 1998.

[47] Ghezzi, C. and Vigna, G. "Mobile Code Paradigms and Technologies: A Case Study" in Mobile Agents, Lecture Notes in Computer Science 1219, Rothermel, K., Popescu-Zeletin, R. Springer-Verlag, Berlin. 1997.

[48] Gong, L and Schemers, R. "Signing, Sealing and Guarding Java Objects" in Mobile Agents and Security, Lecture Notes in Computer Science 1419, Vigna, G (Ed.), Spinger-Verlag, Berlin, 1998.

[49] "An Introduction to LTE". 3GPP LTE Encyclopedia, http://sites.google.com/site/lteencyclopedia/home, Retrieved May 2011.

[50] "Mobile telecommunications standards". Wikipedia. http://en.wikipedia.org/wiki/Template:Mobile_telecommunications_standards. Retrieved May 2011.

[51] "Long Term Evolution (LTE): A Technical Overview". Motorola. 2007. http://www.motorola.com/web/Business/Solutions/Industry%20Solutions/Service%20Providers/Wireless%20Operators/LTE/_Document/Static%20Files/6834_MotDoc_New.pdf. Retrieved May 2011.

[52] Verizon's LTE Network Launching in 30 NFL Cities by End of the Year. http://www.slashgear.com/verizons-lte-network-launching-in-30-nfl-cities-by-end-of-the-year-16102731/. 16-09-2010. Retrieved May 2011.

[53] "Live from Verizon's CES 2011 4G LTE press conference". Engadget. http://www.engadget.com/2011/01/06/live-from-verizons-ces-2011-4g-lte-press-conference/?sort=oldest&refresh=0, 06-01-2011. Retrieved May 2011.

[54] "Verizon promises LTE in 147 markets by the end of 2011". http://www.phonearena.com/news/Verizon-promises-LTE-in-147-markets-by-the-end-of-2011_id17620, 23-03-2011. Retrieved May 2011.

[55] Stefan Parkvall, Erik Dahlman, Anders Furuskär et al; Ericsson, Robert Syputa, Maravedis; "International Telecommunications Union, IMT-Advanced global platform standar; LTE Advanced - Evolving LTE towards IMT-Advanced; Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th 21-24 Sept. 2008.

[56] 3GPP Technical Report: Feasibility study for Further Advancements for E-UTRA (LTE Advanced) http://www.3gpp.org/ftp/Specs/html-info/36912.htm, Retrieved May 2011.

[57] Koshiro Mitsuya et al. Implementation and Evaluation of Dual Stack Mobile IPv6, Keio University, Japan. 2007.

[58] H. Soliman et al. Mobile IPv6 Support for Dual Stack Hosts and Routers. Request for Comments: 5555. June 2009

[59] J. Arkko. Issues in Protecting MIPv6 Binding Updates. Internet Draft draft-arkko-mIPv6-bu-security-01.txt. Work In Progress, IETF, November 2001.

[60] J. Arkko. Security Framework for Mobile IPv6 Route Optimization. Internet Draft draft-arkko-mIPv6ro-secframework-00.txt. Work In Progress, IETF, November 2001.

[61] J. Arkko and P. Nikander. Weak Authentication: How to Authenticate Unknown Principals without Trusted Parties. To appear in Proceedings of Security Protocols Workshop 2002, Cambridge, UK, April 16-19, 2002.

[62] A.Georgiades et al, Trinity Protocol for the authentication of binding updates in mobile IPv6, WSEAS Transactions on communications, Issue 3, Volume 3, July 2004.

Appendix A. Submitted Papers

Trinity Protocol for Authentication of Binding Updates in Mobile IPv6

ANDREW GEORGIADES,
DR YUAN LUO,
DR ABOUBAKER LASEBAE,
PROF. RICHARD COMLEY
Department of Computing Science
Middlesex University
Tottenham campus, White Hart Lane, London, N17 8HR
UNITED KINGDOM
A.georgiades@mdx.ac.uk www.cs.mdx.ac.uk

Abstract: - Improvements have been seen with a jump from 2G to 3G in terms of security. Yet security issues persist and will continue to plague mobile communications into the leap to 4G if not addressed. 4G will use an architecture known as mobile IP. One particular security issue involves the route optimisation technique, which deals with binding updates. This paper examines the variety of security vulnerabilities with binding updates, which include the interception, eavesdropping and modification of data packets, which also include address spoofing, redirection and denial of service attacks. There are a variety of security solutions to prevent these attacks from occurring. Two of the main solutions are cryptography and authentication. These are examined to discover why current security measures are not suitable for the mobile IP environment. Most of the vulnerabilities occur if the attacker knows the IPv6 addresses of the mobile's home agent and the corresponding node and so location privacy is examined as a preventive measure for binding updates. Finally a security solution is proposed called the Trinity protocol. This attempts to solve the security vulnerabilities of binding updates and at the same time over come the short falls of other security solutions.

Key-Words: - Mobile IPv6, Binding Updates, Security, Authentication, Location Privacy, Trinity Protocol.

1 Introduction

Networking has always been vulnerable to a variety of attacks and the next generation of mobile communications is no different. 4G will be based on the transmission of Internet packets only, using an architecture known as mobile IP. This will feature many advantages however security is still a fundamental issue to be resolved. One particular security issue involves the route optimisation technique, which deals with binding updates [4]. This allows the corresponding node to by-pass the home agent router and communicates directly with the mobile node. The home agent has a static address while the mobile node's address changes every time it moves to a new location with a new point of attachment. The home agent keeps track of the mobile node's current address, so that if a correspondent does not know it, it may send the packets to the home address, which will forward the packets to the mobile node [12]. By bypassing the home address with the binding update route optimisation option, the speed of the delivery of packets increases. There are a variety of security vulnerabilities with binding updates [5]. By looking at each of these vulnerabilities, a detailed picture can be constructed of the weaknesses in the current mobile IP architecture leading to an understanding of which security solutions need to be applied and where. Also, many of the different attacks may be possible because of a single or common vulnerability and this must be addressed.

To solve the vulnerability issues of binding updates, a variety of security threats and solutions will be investigated in an attempt to create a unique security solution with the advantages of the previous security solutions yet without any of their disadvantages or drawbacks. Numerous security solutions have been proposed and in chapter four each will be investigated and have their advantages and disadvantages explored.

Distributed Authentication Protocol for the security of Binding Updates in Mobile IPv6

ANDREW GEORGIADES,
DR YUAN LUO,
DR ABOUBAKER LASEBAE,
PROF. RICHARD COMLEY
Department of Computing Science
Middlesex University
Tottenham campus, White Hart Lane, London, N17 8HR
UNITED KINGDOM
A.georgiades@mdx.ac.uk www.cs.mdx.ac.uk

Abstract: - Improvements have been seen with a jump from 2G to 3G in terms of security. Yet security issues persist and will continue to plague mobile communications into the leap to 4G if not addressed. 4G will use an architecture known as mobile IP. One particular security issue involves the route optimisation technique, which deals with binding updates. This paper introduces a distributed authentication protocol. It attempts to solve the security vulnerabilities of binding updates and at the same time over come the short falls of other security solutions by using currently existing technology without introducing any new hardware. The introduction of distributed authentication is aimed at alleviating mobile devices from processor intensive calculations while maintaining a high level of security.

Key-Words: - Mobile IPv6, Binding Updates, Security, Authentication, Location Privacy, Distributed protocol.

1 Introduction

Networking has always been vulnerable to a variety of attacks and the next generation of mobile communications is no different. 4G will be based on the transmission of Internet packets only, using an architecture known as mobile IP. This will feature many advantages however security is still a fundamental issue to be resolved. One particular security issue involves the route optimisation technique, which deals with binding updates [1]. This allows the corresponding node to by-pass the home agent router and communicates directly with the mobile node. The home agent has a static address while the mobile node's address changes every time it moves to a new location with a new point of attachment. The home agent keeps track of the mobile node's current address, so that if a correspondent does not know it, it may send the packets to the home address, which will forward the packets to the mobile node [2]. By bypassing the home address with the binding update route optimisation option, the speed of the delivery of packets increases. There are a variety of security vulnerabilities with binding updates [3].

2 Current Security Solutions

Numerous security solutions have been proposed and each have their advantages and disadvantages. The two main types of security are encryption and authentication. Encryption protects the confidentiality of the data and authentication allows users to verify that they are communicating with validated participants. Different authentication systems exist, such as Kerberos [4] that perform authentication by referring to a central authentication database to compare users credentials.

Other security components include hashes [5], digital signatures [6], address based keys [7] and cryptographically generated addresses [8].

More elaborate systems such as IPSEC [9] and RADIUS [10] based on AAA Authentication, authorization and accounting [11], require the utilization of a central authentication authority. These techniques may not be practical for a mobile environment, and could effectively reduce the users quality of service.

Security protocols, which have been specifically designed for the protection of binding

Binding Update security for Mobile IPv6 using a Distributed Authentication Protocol

ANDREW GEORGIADES,
DR YUAN LUO,
DR ABOUBAKER LASEBAE,
PROF. RICHARD COMLEY
Department of Computing Science
Middlesex University
Tottenham campus, White Hart Lane, London, N17 8HR
UNITED KINGDOM
A.georgiades@mdx.ac.uk www.cs.mdx.ac.uk

Abstract: - Mobile IP architecture will be the basis for the future fourth generation communication technology 4G. However there are potential security issues, which could plague the system if not addressed. One particular security issue involves the route optimisation technique, which deals with binding updates. This paper attempts to solve the security vulnerabilities of binding updates by introducing a distributed authentication protocol. Its primary goal is to over come the short falls of other security solutions by using currently existing technology without introducing any new hardware. Two versions of the protocol are discussed, mobile to mobile and mobile to static node communication both aiming to alleviate mobile devices from processor intensive calculations while maintaining a high level of security. Finally the paper discusses how the proposed protocol addresses the security vulnerabilities.

Key-Words: - Mobile IPv6, Binding Updates, Security, Authentication, Location Privacy, Distributed protocol.

1 Introduction

Networking has always been vulnerable to a variety of attacks and the next generation of mobile communications is no different. 4G will be based on the transmission of Internet packets only, using an architecture known as mobile IP. This will feature many advantages however security is still a fundamental issue to be resolved. One particular security issue involves the route optimisation technique, which deals with binding updates [1]. This allows the corresponding node to by-pass the home agent router and communicates directly with the mobile node. The home agent has a static address while the mobile node's address changes every time it moves to a new location with a new point of attachment. The home agent keeps track of the mobile node's current address, so that if a correspondent does not know it, it may send the packets to the home address, which will forward the packets to the mobile node [2]. By bypassing the home address with the binding update route optimisation option, the speed of the delivery of packets increases. There are a variety of security vulnerabilities with binding updates [3].

2 Current Security Solutions

Numerous security solutions have been proposed and each have their advantages and disadvantages. The two main types of security are encryption and authentication. Encryption protects the confidentiality of the data and authentication allows users to verify that they are communicating with validated participants. Different authentication systems exist, such as Kerberos [4] that perform authentication by referring to a central authentication database to compare users credentials.

Other security components include hashes [5], digital signatures [6], address based keys [7] and cryptographically generated addresses [8].

More elaborate systems such as IPSEC [9] and RADIUS [10] based on AAA Authentication, authorization and accounting [11], require the utilization of a central authentication authority. These techniques may not be practical for a mobile environment, and could effectively reduce the users quality of service.

Security protocols, which have been specifically designed for the protection of binding

Dual Identity Return Routability for the Security of Mobile Ipv6 Binding Updates within the Distributed Authentication Protocol

ANDREW GEORGIADES,
DR YUAN LUO,
DR ABOUBAKER LASEBAE,
PROF. RICHARD COMLEY
Department of Computing Science
Middlesex University
Hendon campus, The Burroughs, London, NW4 4BT
UNITED KINGDOM
A.georgiades@mdx.ac.uk www.cs.mdx.ac.uk

Abstract: - The future fourth generation 4G networks will provide us with a paradigm shift in how mobile telecommunications will operate. It will be solely based on packet switching using mobile IPv6. However binding update route optimisation is vulnerable to a variety of security attacks. This paper attempts to reduce the security vulnerabilities by creating a new security protocol by first investigating the possible future technologies which may be incorporated into 4G mobile phones. Various technologies such as WI-FI and WiMax will be looked at but one in particular may be of particular interest, sim cards which allow the user to have multiple phone numbers. Using this technology and combining it with the established security protocol return routability, a new enhanced security solution is created called Dual Identity Return Routability. This solution provides an enhanced reachability test and a cheap authentication method, which can be incorporated into the distributed authentication protocol or be used as a stand-alone solution.

Key-Words: - Mobile IPv6, Binding Updates, Security, Authentication, Return Routability, Dual Identity.

1 Introduction

Before a security solution can be designed for a future telecommunication network, it is wise and vital to take a look at the emerging technologies and economical factors, which may impact the very core of the telecommunications industry, as we know it. This paper will present some predictions of which technologies will be incorporated into the Forth Generation of mobile telecommunications, technologies, which may have such a fundamental impact, that it will create a paradigm shift in the way the service is run. Only then can the network architecture be understood and a security solution crafted to adequately take advantage of its environment. This paper attempts to find a solution to prevent binding updates in Mobile Ipv6 from being susceptible to masquerading and impersonation attacks.

2 Problem Definition

Mobile IP has primarily been designed for the ease of mobility of communicating devices. It is the underlining architecture for the fourth generation of mobile phones. Due to the nature of TCP/IP, only static IP addresses are permitted to be used within the network. This causes problems for mobile nodes, which wish to migrate to a new location yet still remain connected to the network. This is because physically moving to another location results in a new attachment to a wireless network node and as a result the IP address would change. Mobile IP solves this issue by employing two addresses [1].

The First address belongs to the home agent, which acts as a proxy for the mobile node and ensures the mobile node remains reachable by having a static address.

The mobile node itself has a dynamic address and this changes every time the node is associated with another point of attachment. Each time the mobile node migrates to a new location, it is assigned a new IP address and the home agent is informed of that new address. A node wishing to contact the mobile node must contact the home agent, which will tunnel the data packets to the current address of the mobile node. Correspondent nodes communicate by sending packets to the

Distributed Authentication Protocol Utilizing Dual Identity Return Routability for the Security of Binding Updates within Mobile IPv6

ANDREW GEORGIADES,
DR YUAN LUO,
DR ABOUBAKER LASEBAE,
PROF. RICHARD COMLEY
Department of Computing Science
Middlesex University
Hendon campus, The Burroughs, London, NW4 4BT
UNITED KINGDOM
A.georgiades@mdx.ac.uk www.cs.mdx.ac.uk

Abstract: - The future fourth generation 4G networks will provide us with a paradigm shift in how mobile telecommunications will operate. It will be solely based on packet switching using mobile IPv6. However binding update route optimisation is vulnerable to a variety of security attacks. This paper attempts to reduce the security vulnerabilities by creating a new security protocol by first investigating the possible future technologies which may be incorporated into 4G mobile phones. Various technologies such as WI-FI and WiMax will be looked at but one in particular may be of particular interest, sim cards which allow the user to have multiple phone numbers. Using this technology and combining it with the established security protocol return routability, a new enhanced security solution is created called Dual Identity Return Routability. This solution provides an enhanced reachability test and a cheap authentication method, which can be incorporated into the distributed authentication protocol, which is demonstrated, or be used as a stand-alone solution.

Key-Words: - Mobile IPv6, Binding Updates, Security, Authentication, Return Routability, Dual Identity.

1 Introduction

Before a security solution can be designed for a future telecommunication network, it is wise and vital to take a look at the emerging technologies and economical factors, which may impact the very core of the telecommunications industry, as we know it. This paper will present some predictions of which technologies will be incorporated into the Forth Generation of mobile telecommunications, technologies, which may have such a fundamental impact, that it will create a paradigm shift in the way the service is run. Only then can the network architecture be understood and a security solution crafted to adequately take advantage of its environment. This paper attempts to find a solution to prevent binding updates in Mobile Ipv6 from being susceptible to masquerading and impersonation attacks.

2 Problem Definition

Mobile IP has primarily been designed for the ease of mobility of communicating devices. It is the underlining architecture for the fourth generation of mobile phones. Due to the nature of TCP/IP, only static IP addresses are permitted to be used within the network. This causes problems for mobile nodes, which wish to migrate to a new location yet still remain connected to the network. This is because physically moving to another location results in a new attachment to a wireless network node and as a result the IP address would change. Mobile IP solves this issue by employing two addresses [1].

The First address belongs to the home agent, which acts as a proxy for the mobile node and ensures the mobile node remains reachable by having a static address.

The mobile node itself has a dynamic address and this changes every time the node is associated with another point of attachment. Each time the mobile node migrates to a new location, it is assigned a new IP address and the home agent is informed of that new address. A node wishing to contact the mobile node must contact the home agent, which will tunnel the data packets to the current address of the mobile node. Correspondent nodes communicate by sending packets to the

Location Privacy in Mobile IPv6 Distributed Authentication Protocol Using Mobile Home Agents

ANDREWGEORGIADES
DR YUAN LUO
DR ABOUBAKER LASEBAE
PROF. RICHARD COMLEY
Department of Computer Science
Middlesex University
Hendon Campus, The Burroughs, London, NW4 4BT
UNITED KINGDOM
Andrew_georgiades@yahoo.co.uk http://www.andrewgeorgiades.com

Abstract: - Mobile IPv6 will be the basis for the fourth generation 4G networks which will completely revolutionize the way telecommunication devices operate. This paradigm shift will occur due to the sole use of packed switching networks. Mobile IPv6 utilizes binding updates as a route optimization to reduced triangle routing between the mobile node, the home agent and the correspondent node, allowing direct communication between the mobile node and the correspondent. However, direct communication between the nodes produces a range of security vulnerabilities, which the home agent avoided. This paper attempts to provide the advantages of using the home agent as an intermediary whilst reducing the latency of triangle routing. This can be achieved with the proposed use of a mobile home agent which essentially follows the mobile node as it moves between points of attachment providing location privacy and pseudo-direct communication, which can be incorporated into the distributed authentication protocol or be used as a stand alone solution.

Key-Words: - Mobile Home Agent, MIPv6, Distributed Authentication Protocol, 4G, Location Privacy

1 Introduction

Mobile IPv6 is the next step in the evolution of networking. The most widely used internet protocol are currently networks based on IPv4 which are restricted to 32 bit addresses. This provides a number of IP addresses which, over time, has became limited to the number of devices which need them. Network address translation has helped to delay the need for more address. However a new Internet protocol was inevitably created to solve this issue, IPv6. Ipv6 addresses are 128 bit providing 3.4×10^{38} address which solves the issue of address limitation however as most devices are becoming mobile, IPv6 provides no method for them to migrate to a new location as the IP addresses are static [13].

Mobile IPv6 solves this issue by providing an infrastructure which allows the mobile node to acquire a new address every time it moves to a new point of attachment and yet still remain reachable as it has a home agent which has an IP address which remains static and also keeps track of the mobile node's current location. The home agent is the first point of contact when attempting to contact the mobile node as the home agent acts as a proxy and tunnels messages to the mobile node. This is called triangle routing and the latency of communication between the nodes increases the further away the mobile node travels from the home agent [8].

The introduction of the route optimization protocol allows the mobile node to communicate directly with its correspondents with the use of binding updates. However these are vulnerable to a variety of attacks such as interception, modification, impersonation and redirection. Binding updates are also susceptible to denial of service attacks.

However, several security solutions have been created which attempt to protect the binding updates, such as CAM [11] and the distributed authentication protocol [5]. But non of these address the issue of location privacy, for if the attacker is unable to determine the location of the mobile node, he will not be able to attack it.

This paper will look at the advantages and disadvantages of current location privacy security solutions in Mobile IPv6. It will then look at the new technology of mobile autonomous software agents, which can exist and move independently within heterogeneous networks. The paper will then go on to suggest that mobile agents can be used in a security solution where they will act as mobile home agents providing location privacy without increasing

Introducing Mobile Home Agents into the Distributed Authentication Protocol to Achieve Location Privacy in Mobile IPv6

Andrew Georgiades, Dr Yuan Luo, Dr Aboubaker Lasebae, Prof. Richard Comley

***Abstract*—Mobile IPv6 will be the basis for the fourth generation 4G networks which will completely revolutionize the way telecommunication devices operate. This paradigm shift will occur due to the sole use of packed switching networks. Mobile IPv6 utilizes binding updates as a route optimization to reduced triangle routing between the mobile node, the home agent and the correspondent node, allowing direct communication between the mobile node and the correspondent. However, direct communication between the nodes produces a range of security vulnerabilities, which the home agent avoided. This paper attempts to provide the advantages of using the home agent as an intermediary whilst reducing the latency of triangle routing. This can be achieved with the proposed use of a mobile home agent which essentially follows the mobile node as it moves between points of attachment providing location privacy and pseudo-direct communication, which can be incorporated into the distributed authentication protocol or be used as a stand alone solution.**

***Keywords*—Mobile Home Agent, MIPv6, Distributed Authentication Protocol, 4G, Location Privacy.**

I. Introduction

Mobile IPv6 is the next step in the evolution of networking. The most widely used internet protocols are ones currently based on IPv4 networks which are restricted to 32 bit addresses. This provides a finite number of IP addresses which, over time, has become limited to the number of devices which need them. Network address translation has helped to delay the need for more address. However a new Internet protocol was inevitably created to solve this issue, IPv6. IPv6 addresses are 128 bit providing 3.4×1038 addresses which solves the issue of address limitation however as most devices are becoming mobile, IPv6 provides no method for them to migrate to a new location as the IP addresses are static [1].

Mobile IPv6 solves this issue by providing an infrastructure which allows the mobile node to acquire a new address every time it moves to a new point of attachment and yet still remain reachable as it has a home agent which has an IP address which remains static and also keeps track of the mobile node's current location. The home agent is the first point of contact when attempting to contact the mobile node as the home agent acts as a proxy and tunnels messages to the mobile node. This is called triangle routing and the latency of communication between the nodes increases the further away the mobile node travels from the home agent [2].

The introduction of the route optimization protocol allows the mobile node to communicate directly with its correspondents with the use of binding updates. However these are vulnerable to a variety of attacks such as interception, modification, impersonation and redirection. Binding updates are also susceptible to denial of service attacks.

However, several security solutions have been created which attempt to protect the binding updates, such as CAM [3] and the distributed authentication protocol [4]. But non of these address the issue of location privacy, for if the attacker is unable to determine the location of the mobile node, he will not be able to attack it.

This paper will look at the advantages and disadvantages of current location privacy security solutions in Mobile IPv6. It will then look at the new technology of mobile autonomous software agents, which can exist and move independently within heterogeneous networks. The paper will then go on to suggest that mobile agents can be used in a security solution where they will act as mobile home agents providing location privacy without increasing communication latency. This solution can be used as a stand alone solution or be used as part of the distributed authentication protocol.

Manuscript received February, 2009: This work is sponsored by Middlesex University, London, England and is part of the research undertaken by the PhD candidate Andrew Georgiades which goes towards achieving his doctorate.

Andrew Georgiades, PhD candidate, Middlesex University, and Production Innovation Coordinator, BBC, Television Centre, London, England. He works for the Innovation department of the British Broadcasting Corporation, Providing support to and introducing new and innovative technologies into the BBC's top Entertainment, Comedy and Event productions. (Andrew_georgiades@yahoo.co.uk).

Dr Yuan Luo, Senior Lecturer and Program leader of BEng Computer Communications and Networks, Middlesex University, London, England. (y.luo@mdx.ac.uk)

Dr Abubaker Lasebae, Principle Lecturer and Director of post graduate programs (Computer Communications), Program leader MSc Computer and Network Security, Middlesex University, London, England. (a.lasebae@mdx.ac.uk)

Professor Richard Comley, Associate Dean – Research, Professor of Computer Communications, Middlesex University, London, England. (r.comley@mdx.ac.uk)

Appendix B. Simulation Source Code

```
//
// This file is part of a PhD OMNeT++ simulation experiment.
//
// Copyright (C) 2010 Andrew Georgiades
//
//
// This file is distributed WITHOUT ANY WARRANTY.
//
//
simple Node
{
    parameters:
        @display("i=block/routing");
    gates:
        inout gate[];
}
simple MN extends Node
{
    parameters:
        @display("i=device/pocketpc");
}
simple CN extends Node
{
    parameters:
        @display("i=device/server");
}
simple HA extends Node
{
    parameters:
        @display("i=device/server2");
}
simple Router extends Node
{
    parameters:
        @display("i=abstract/router");
}
simple Attacker extends Node
{
    parameters:
        @display("i=device/wifilaptop");
}
simple MHA extends Node
{
    parameters:
        @display("i=block/app");
}
simple PoA extends Node
{
    parameters:
        @display("i=device/antennatower");
}
simple AccessPoint extends Node
{
    parameters:
        @display("i=device/accesspoint");
}
//
// Same as Tictoc12
//
network Mipv6
{
    @display("bgb=736,476");
    types:
        channel Channel extends ned.DelayChannel
        {
            delay = 100ms;
        }
        channel Ethernet extends ned.DelayChannel
        {
            delay = 100ms;
        }
```

```
    channel Wireless4G extends ned.DelayChannel
    {
        delay = 100ms;
    }
    channel WirelessWiFi extends ned.DelayChannel
    {
        delay = 100ms;
    }
    channel fiberline extends ned.DatarateChannel
    {
        parameters:
            delay = 1us;
            datarate = 512*1000000;
    }

    channel ethernetline extends ned.DatarateChannel
    {
        parameters:
            delay = 0.1us;
    }
submodules:
    // tic[6]: Node;
    MN: MN {
        @display("p=315,32");
    }
    CN: CN {
        @display("p=371,400");
    }
    //HA[6]: HA;
    Attacker: Attacker {
        @display("p=286,145");
    }
    //MHA[0]: MHA;
    HA: HA {
        @display("p=91,190");
    }
    Router1: Router {
        @display("p=205,295");
    }
    Router2: Router {
        @display("p=465,212");
    }
    PoA: PoA {
        @display("p=77,64");
    }
    AccessPoint: AccessPoint {
        @display("p=529,144");
    }
    CHA: HA {
        @display("p=510,344");
    }
connections:
    //tic[0].gate++ <--> Channel <--> tic[1].gate++;
    //tic[1].gate++ <--> Channel <--> tic[2].gate++;
    //tic[1].gate++ <--> Channel <--> tic[4].gate++;
    //tic[3].gate++ <--> Channel <--> tic[4].gate++;
    //tic[4].gate++ <--> Channel <--> tic[5].gate++;
    Router1.gate++ <--> Channel <--> Router2.gate++;
    Router2.gate++ <--> Channel <--> AccessPoint.gate++;

    HA.gate++ <--> Channel <--> Router1.gate++;
    HA.gate++ <--> Channel <--> PoA.gate++;
    MN.gate++ <--> Channel <--> PoA.gate++;
    Attacker.gate++ <--> Channel <--> MN.gate++;
    Attacker.gate++ <--> Channel <--> PoA.gate++;
    Attacker.gate++ <--> Channel <--> HA.gate++;
    Attacker.gate++ <--> Channel <--> Router1.gate++;
    Attacker.gate++ <--> Channel <--> CN.gate++;
    Attacker.gate++ <--> Channel <--> Router2.gate++;
    Attacker.gate++ <--> Channel <--> AccessPoint.gate++;
    Attacker.gate++ <--> Channel <--> CHA.gate++;
    Router1.gate++ <--> Channel <--> CN.gate++;
    Router1.gate++ <--> Channel <--> CHA.gate++;
    CHA.gate++ <--> Channel <--> CN.gate++;
    MN.gate++ <--> Channel <--> AccessPoint.gate++;
```

```
//MN[0].gate++ <--> Channel <--> MN[1].gate++;
//MN[1].gate++ <--> Channel <--> MN[2].gate++;

    //MN[0].gate++ <--> Channel <--> MN[1].gate++;
    //MN[1].gate++ <--> Channel <--> MN[2].gate++;
    //MN[1].gate++ <--> Channel <--> MN[4].gate++;
    //MN[3].gate++ <--> Channel <--> MN[4].gate++;
    //MN[4].gate++ <--> Channel <--> MN[5].gate++;
}
```

```
//
// This file is part of a PhD OMNeT++ simulation experiment.
//
// Copyright (C) 2010 Andrew Georgiades
//
//
// This file is distributed WITHOUT ANY WARRANTY.
//
//

#include <stdio.h>
#include <string.h>
#include <omnetpp.h>
#include "tictoc15_m.h"


class Node : public cSimpleModule
{
  private:
    long numSent;
    long numReceived;
    cLongHistogram hopCountStats;
    cOutVector hopCountVector;
    int current;
    int GateID;
    int src;
    int dest;
    int counter;
    int counterA;
    int home;
    int mobile;
    int MN;
    int CN;
    int attack;
    int directAttack;
    int PoA;
    int HA;
    int Router1;
    int CHA;
    int Router2;
    int accesspoint;
    int attacker;
    int rr;
    int dirr1;
    int dirr2;
    int cga;
    int cgahome;
    int packetsource;
    int lostConnection;
    int packetCount;
    simtime_t timeout;  // timeout
    cMessage *timeoutEvent;  // holds pointer to the timeout self-message
    int cgasecurity;
    int attackercga;
    int attackercgaaddress;
    int CoT;
    int HoT;
    int CNHABuffer;
    int rrAttack;
    int dap;
    int dapMN;
    int dapHA;
    int dirr;
    int MNdirr;
    int dirrCoT;
    int dirrHoT;
    int MHA;
    //int dapMNCode;
    //int dapHACode;

  public:
                Node();
                virtual ~Node();
                int dapMNCode;
                int dapHACode;
```

```
        int dapATCode;

  protected:
    virtual TicTocMsg15 *generateMessage();
    //virtual TicTocMsg15 *lostMessage();
    virtual TicTocMsg15 *generateBU();
    virtual TicTocMsg15 *generateBA();
    virtual TicTocMsg15 *generateCoT();
    virtual TicTocMsg15 *generateHoT();
    virtual TicTocMsg15 *generatedirrCoT();
    virtual TicTocMsg15 *generatedirrHoT();
    virtual TicTocMsg15 *generatePackets();
    virtual TicTocMsg15 *tunnelMg();
    virtual TicTocMsg15 *updatingHA();
    virtual TicTocMsg15 *attackMsg();
    virtual TicTocMsg15 *attackST();
    virtual TicTocMsg15 *generateKey();
    virtual TicTocMsg15 *requestdapMN();
    virtual TicTocMsg15 *requestdapHA();
    virtual TicTocMsg15 *dapMNauth();
    virtual TicTocMsg15 *dapHAauth();
    virtual void forwardMessage(TicTocMsg15 *msg);
    virtual void initialize();
    virtual void handleMessage(cMessage *msg);
    //virtual void handleMissing(cMessage *msg);

    // The finish() function is called by OMNeT++ at the end of the simulation:
    virtual void finish();
};

Define_Module(Node);


Node::Node()
{
    timeoutEvent = NULL;
}

Node::~Node()
{
    cancelAndDelete(timeoutEvent);
}

void Node::initialize()
{
    // Initialize variables
    numSent = 0;
    numReceived = 0;
    WATCH(numSent);
    WATCH(numReceived);

    hopCountStats.setName("hopCountStats");
    hopCountStats.setRangeAutoUpper(0, 10, 1.5);
    hopCountVector.setName("HopCount");
    counter=0;
    counterA=0;

    home=0; //1 = MN node on home network. 0 = MN on foreign network
    mobile=0; //0 = CN is static. 1 = CN is mobile with it's own HA.

    attack=1; // 0 = Attack is dormant. 1 = Attack will send a false BU.
    directAttack=1;

    cgasecurity=1;//turn on CGA = 1, turn off CGA=0;
    rr=1; //0 return routability off. 1 = on.
    dap=1;//0 distributed authentication protocol off. 1 = on.
    dirr=1;//rr needs to be on to use this.
    MHA=1;//1 turns on Mobile home agent. 0 is off. To use this home must =1.

    attackercga=1; //0 for non, 1 for it's own CGA or 2 to try to spoof MN/HA CGA address
    rrAttack=2;//1 use HA as HoT destination. 2 Spoof HA with Attacker address.



    dapMN=0;
```

```
dapHA=0;
PoA=2;
HA=3;
Router1=4;
CHA=6;
Router2=7;
accesspoint=8;
attacker=9;
MNdirr=10;

dirr1=12;
dirr2=13;
cga=101;
cgahome=103;
attackercgaaddress=109;
lostConnection=0;
packetCount=0;
timeout = 2.0;
timeoutEvent = new cMessage("timeoutEvent");
CoT=0;
HoT=0;
dirrCoT=0;
dirrHoT=0;
CNHABuffer=0;
dapMNCode=123;
dapHACode=123;
dapATCode=321;


if((attackercga==1)||(attackercga==2))
{
        attacker=attackercgaaddress;
}


if(cgasecurity==1)
{

        MN=cga;
        HA=cgahome;
        if(MHA==1)
        {
        accesspoint=cgahome;
        }
        }
else
{



if(home==1)
{
        MN=1;
}
else if(home==0)
{
        MN=10;
}
}//end cga security

if(mobile==0)
   {
        CN=5;
   }
   else if(mobile==1)
   {
        CN=11;
   }
// Module 0 sends the first message
//if (getIndex()==0  )
//{
   // Boot the process scheduling the initial message as a self-message.
  //  TicTocMsg15 *msg = generateMessage();
  //  scheduleAt(0.0, msg);
//}
```

```
  if (strcmp("MN", getName()) == 0)
    {
      // create and send first message on gate "out". "tictocMsg" is an
      // arbitrary string which will be the name of the message object.
      //cMessage *msg = new cMessage("tictocMsg");
      //send(msg, "out");

          if(cgasecurity==0)
          {

          TicTocMsg15 *msg = generateMessage();
            //TicTocMsg15 *msg2 = generateMessage();
               scheduleAt(0.0, msg);
               //scheduleAt(simTime()+1.0, msg2);
               scheduleAt(simTime()+timeout, timeoutEvent);
          }
          else if(cgasecurity==1)
          {
                              TicTocMsg15 *msg = generateMessage();
                                    //TicTocMsg15 *msg2 = generateMessage();
                                       scheduleAt(0.0, msg);
                                       //scheduleAt(simTime()+1.0, msg2);
                                       scheduleAt(simTime()+timeout, timeoutEvent);
          }
    }
  if (strcmp("Attacker", getName()) == 0)
  {

          if(attack==1)
          {
                        TicTocMsg15 *attackmsg = generateMessage();
               EV << attackmsg << endl;
               scheduleAt(0.0, attackmsg);
               scheduleAt(simTime()+timeout, timeoutEvent);
          }
  }

}

void Node::handleMessage(cMessage *msg)
{
   //TicTocMsg15 *ttmsg = check_and_cast<TicTocMsg15 *>(msg);

  /* if ((msg->isSelfMessage())&&(simTime()>0.5))
   {
          if(lostConnection>5)
          {
          lostConnection=0;

          }
          else
          {
                    cancelEvent(msg);
                    TicTocMsg15 *msg3 = generateMessage();
                    scheduleAt(simTime()+1.0, msg3);
          }
   }
*/



  if (msg==timeoutEvent)
    {
      // If we receive the timeout event, that means the packet hasn't
      // arrived in time and we have to re-send it.
      EV << "Timeout expired, resending message and restarting timer\n";
     // cMessage *msg = new cMessage("");
      //send(msg, "out");
      //send(msg, "out");
           // TicTocMsg15 *msg = generateMessage();
      bubble("Timeout expired, resending message and restarting timer!");
      EV << "Generating another message: ";
                        TicTocMsg15 *newmsg = generateMessage();
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
```

```
                numSent++;
        scheduleAt(simTime()+timeout, timeoutEvent);
    }
    else // message arrived
    {
        TicTocMsg15 *ttmsg = check_and_cast<TicTocMsg15 *>(msg);

        if ((((strcmp(ttmsg->getName(),"Acknowledgement")==0)&&((strcmp("MN", getName()) == 0)))||
                ((strcmp(ttmsg->getName(),"BU")==0)&&((strcmp("MN", getName()) == 0)))||
                (((strcmp(ttmsg->getName(),"CoT")==0)&&((strcmp("MN", getName()) == 0))))||
                (((strcmp(ttmsg->getName(),"requestdapMN")==0)&&((strcmp("MN", getName()) == 0))))||
                (((strcmp(ttmsg->getName(),"dapMNauth")==0)&&((strcmp("MN", getName()) == 0))))||
                (((strcmp(ttmsg->getName(),"BA")==0)&&((strcmp("MN", getName()) == 0))))||
                (((strcmp(ttmsg->getName(),"Acknowledgement")==0)&&((strcmp("MN", getName()) ==
0))))||
                (((strcmp(ttmsg->getName(),"PacketRequest")==0)&&((strcmp("MN", getName()) == 0))))||
                (((strcmp(ttmsg->getName(),"PacketReply")==0)&&((strcmp("MN", getName()) == 0)))))||

                (((strcmp(ttmsg->getName(),"Acknowledgement")==0)&&((strcmp("Attacker", getName())
== 0)))||
                ((strcmp(ttmsg->getName(),"BU")==0)&&((strcmp("Attacker", getName()) == 0)))||
                (((strcmp(ttmsg->getName(),"CoT")==0)&&((strcmp("Attacker", getName()) == 0))))||
                (((strcmp(ttmsg->getName(),"requestdapMN")==0)&&((strcmp("Attacker", getName()) ==
0))))||
                (((strcmp(ttmsg->getName(),"dapMNauth")==0)&&((strcmp("Attacker", getName()) ==
0))))||
                (((strcmp(ttmsg->getName(),"BA")==0)&&((strcmp("Attacker", getName()) == 0))))||
                (((strcmp(ttmsg->getName(),"Acknowledgement")==0)&&((strcmp("Attacker", getName())
== 0))))||
                (((strcmp(ttmsg->getName(),"PacketRequest")==0)&&((strcmp("Attacker", getName()) ==
0))))||
                (((strcmp(ttmsg->getName(),"PacketReply")==0)&&((strcmp("Attacker", getName()) ==
0))))))
                {

            // Acknowledgement received -- delete the stored message and cancel
          // the timeout event.
          EV << "Timer cancelled.\n";
          cancelEvent(timeoutEvent);

          // Ready to send another one.
         // cMessage *msg = new cMessage("tictocMsg");
         // send(msg, "out");
          scheduleAt(simTime()+timeout, timeoutEvent);
                }
     // }


/*
   if(strcmp(msg->getName(),"Acknowledgement")==0)
            {
                bubble("test1!");
            }

            if(strcmp(msg->getName(),"Request")==0)
            {
                bubble("test2!!");
            }
*/

   if ((strcmp("CN", getName()) == 0)&&(mobile==0))
   {
   current=CN;

   }
   else if ((strcmp("CN", getName()) == 0)&&(mobile==1))
     {
     current=CN;

     }
   else if (strcmp("AccessPoint", getName()) == 0)
        {
        current=accesspoint;

        }
```

```
else if ((strcmp("MN", getName()) == 0)&&(home==1))
{

                        if ( ttmsg->getDestination()==MNdirr)
                        {
                                current=MNdirr;
                        }
                        else
                        {
                current=MN;
                        }
}

else if ((strcmp("MN", getName()) == 0)&&(home==0))
   {
                if ( ttmsg->getDestination()==MNdirr)
                                        {
                                                current=MNdirr;
                                        }
                                        else
                                        {
                        current=MN;
                                        }
   }
else if ((strcmp("HA", getName()) == 0)&&(home==1))
   {
                bubble("home=1");
                current=HA;
   }
else if ((strcmp("HA", getName()) == 0)&&(home==0))
      {
                bubble("home=0");
                current=HA;
      }
/* else if ((strcmp("MNdirr", getName()) == 0)&&(dirr==1))
                {
                current=MNdirr;

                }*/
else if (strcmp("Attacker", getName())== 0)
          {
                bubble("attacker here");
                current=attacker;
          }




if (ttmsg->getDestination()==current)
{
   // Message arrived
   int hopcount = ttmsg->getHopCount();
   EV << "Message " << ttmsg << " arrived after " << hopcount << " hops.\n";
   bubble("ARRIVED, starting new one!");

   // update statistics.
   numReceived++;
   hopCountVector.record(hopcount);
   hopCountStats.collect(hopcount);
   packetsource=(ttmsg->getSource());



   if ((strcmp(ttmsg->getName(),"Acknowledgement")==0)&&((strcmp("AccessPoint", getName()) == 0))&&(MHA==1))
         {

           //delete ttmsg;
             bubble("Tunneling to MN");
                                // Generate another one.
                                EV << "Tunneling to MN: ";
                               // TicTocMsg15 *newmsg = tunnelMg();
```

```
                    ttmsg->setDestination(MN);
                    EV << ttmsg << endl;
                    forwardMessage(ttmsg);
                    numSent++;
    }
else if ((strcmp(ttmsg->getName(),"BA")==0)&&((strcmp("AccessPoint", getName()) == 0))&&(MHA==1))
        {

              //delete ttmsg;
          bubble("Tunneling to MN");
                              // Generate another one.
                              EV << "Tunneling to MN: ";
                             // TicTocMsg15 *newmsg = tunnelMg();
                              ttmsg->setDestination(MN);
                              EV << ttmsg << endl;
                              forwardMessage(ttmsg);
                              numSent++;
        }
else if ((strcmp(ttmsg->getName(),"Acknowledgement")==0)&&((strcmp("HA", getName()) == 0)))
{

     //delete ttmsg;
  bubble("Tunneling to MN");
                    // Generate another one.
                    EV << "Tunneling to MN: ";
                   // TicTocMsg15 *newmsg = tunnelMg();
                    ttmsg->setDestination(MN);
                    EV << ttmsg << endl;
                    forwardMessage(ttmsg);
                    numSent++;
}
else if ((strcmp(ttmsg->getName(),"BA")==0)&&((strcmp("HA", getName()) == 0)))
    {

     //delete ttmsg;
      bubble("Tunneling to MN");
                    // Generate another one.
                    EV << "Tunneling to MN: ";
                   // TicTocMsg15 *newmsg = tunnelMg();
                    ttmsg->setDestination(MN);
                    EV << ttmsg << endl;
                    forwardMessage(ttmsg);
                    numSent++;
    }
else if ((strcmp(ttmsg->getName(),"CoA")==0)&&(strcmp("AccessPoint", getName()) == 0)&&(MHA==1))
        {

              MN=ttmsg->getSource();
          delete ttmsg;
            bubble("CoA Updated");

                              // numSent++;
        }
else if ((strcmp(ttmsg->getName(),"CoA")==0)&&((strcmp("HA", getName()) == 0)))
    {

     MN=ttmsg->getSource();
          delete ttmsg;
      bubble("CoA Updated");

                    // numSent++;
    }

else if (strcmp(ttmsg->getName(),"dapMNauth")==0)
            {

     // Generate another one.
     dapMN=1;
     //EV << dapMNCode << endl;
     //EV << dapHACode << endl;
     //dapMNCode=345;
     //EV << dapMNCode << endl;
     if(ttmsg->getSource()==attacker)
     {
                dapMNCode=321;
```

```
        }
        else if(ttmsg->getSource()==MN)
        {
                dapMNCode=123;
        }
        delete ttmsg;

                if((dapHA==1)&&(dapMN==1))
                {
                        if(dapMNCode==dapHACode)
                        {
                        EV << "Authenticated: ";
                        bubble("Authenticated");
                       TicTocMsg15 *newmsg = generateBA();//tunnelMg();
                       //ttmsg->setDestination(MN);
                       EV << newmsg << endl;
                       forwardMessage(newmsg);
                       numSent++;
                       dapHA=0;
                       dapMN=0;
                        }
                        else if(dapMN>=2)
                        {
                                EV << "Attack in Progress. Authentication failed: ";
                                bubble("Attck in progress. Authentication failed");
                                dapHA=0;
                                dapMN=0;

                        }
                        else
                        {
                                EV << "Authentication failed: ";
                                bubble("Authentication failed");
                                dapHA=0;
                                dapMN=0;
                        }
                }
                else{
                        EV << "dapMN Received: ";
                        bubble("dapMN Received");
                }



        }
else if (strcmp(ttmsg->getName(),"dapHAauth")==0)
                {
        //ttmsg->setSource(MN);
                delete ttmsg;
                 // Generate another one.
                dapHA=1;
                //EV << dapMNCode << endl;
                //EV << dapHACode << endl;
                        if((dapHA==1)&&(dapMN==1))
                        {
                                if(dapMNCode==dapHACode)
                                                                {
                                EV << "Authenticated: ";
                                bubble("Authenticated");

                                TicTocMsg15 *newmsg = generateBA();//tunnelMg();
                               //ttmsg->setDestination(MN);
                               EV << newmsg << endl;
                               forwardMessage(newmsg);
                               numSent++;
                               dapHA=0;
                               dapMN=0;
                                                                }
                                else if(dapHA>=2)
                                                                {
                                                                        EV << "Attack in Progress.
Authentication failed: ";
                                                                        bubble("Attck in progress.
Authentication failed");
                                                                        dapHA=0;
```

```
                                                                dapMN=0;

                                                        }
                        else
                        {
                                EV << "Authentication failed: ";
                                bubble("Authentication failed");
                                dapHA=0;
                                dapMN=0;
                        }
                }
                else{
                        EV << "dapHA Received: ";
                        bubble("dapHA Received");
                }



        }
else if((strcmp(msg->getName(),"requestdapMN")==0)&&((strcmp("Attacker", getName()) == 0)))
                {
                delete ttmsg;
//dapMNCode=54321;
                //dapMNCode=dapATCode;
                bubble("requestdapATT Received");
                        // Generate another one.
                        EV << "Sending Auth data: ";
                        TicTocMsg15 *newmsg = dapMNauth();
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
                        numSent++;

                }
else if(strcmp(msg->getName(),"requestdapMN")==0)
                {
                delete ttmsg;
              //dapMNCode=123;
                bubble("requestdapMN Received");
                        // Generate another one.
                        EV << "Sending Auth data: ";
                        TicTocMsg15 *newmsg = dapMNauth();
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
                        numSent++;

                }
else if((strcmp(msg->getName(),"requestdapHA")==0)&&(strcmp("AccessPoint", getName()) == 0)&&(MHA==1))
                        {
                        /*delete ttmsg;

                        bubble("requestdapHA Received");
                                // Generate another one.
                                EV << "Sending Auth data: ";
                                TicTocMsg15 *newmsg = dapHAauth();
                               //  EV << newmsg << endl;
                                forwardMessage(newmsg);
                                numSent++;*/
        bubble("Tunneling to CN");
        //dapHACode=123;
                        // Generate another one.
                        EV << "Tunneling to CN: ";
                       // TicTocMsg15 *newmsg = tunnelMg();
                        ttmsg->setDestination(CN);
                        ttmsg->setSource(accesspoint);
                        ttmsg->setName("dapHAauth");
                        EV << ttmsg << endl;
                        forwardMessage(ttmsg);
                        numSent++;

                        }
else if(strcmp(msg->getName(),"requestdapHA")==0)
                {
                /*delete ttmsg;

                bubble("requestdapHA Received");
```

```
                                        // Generate another one.
                                        EV << "Sending Auth data: ";
                                        TicTocMsg15 *newmsg = dapHAauth();
                                       //  EV << newmsg << endl;
                                        forwardMessage(newmsg);
                                        numSent++;*/
        bubble("Tunneling to CN");
        //dapHACode=123;
                                    // Generate another one.
                                    EV << "Tunneling to MN: ";
                                   // TicTocMsg15 *newmsg = tunnelMg();
                                    ttmsg->setDestination(CN);
                                    ttmsg->setSource(HA);
                                    ttmsg->setName("dapHAauth");
                                    EV << ttmsg << endl;
                                    forwardMessage(ttmsg);
                                    numSent++;

                                }
else if ((strcmp(ttmsg->getName(),"CGA-Request")==0)&&((strcmp("CN", getName()) == 0)))
            {

                    if(ttmsg->getSource()==cgahome)
                    {
                delete ttmsg;
                  bubble("CGA Authenticated");
                  EV << "Generating another message: ";
                                    TicTocMsg15 *newmsg = generateMessage();
                                    EV << newmsg << endl;
                                    forwardMessage(newmsg);
                                    numSent++;
                    }
                    else if((ttmsg->getSource()==attackercgaaddress)&&(attackercga==1))
                                        {
                                    delete ttmsg;
                                      bubble("CGA Authenticated");
                                      EV << "Generating another message: ";
                                                        TicTocMsg15 *newmsg = generateMessage();
                                                        EV << newmsg << endl;
                                                        forwardMessage(newmsg);
                                                        numSent++;
                                        }
                    else if(ttmsg->getSource()!=cgahome)
                    {
                            delete ttmsg;
                                        bubble("Authenticated Failed");
                    }
                                    // numSent++;
            }
        else if (strcmp(msg->getName(),"Acknowledgement")==0)
                    {

        delete ttmsg;

                        // Generate another one.
                        EV << "Generating another message: ";
                        TicTocMsg15 *newmsg = generateBU();//tunnelMg();
                        //ttmsg->setDestination(MN);
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
                        numSent++;
                        HoT=0;
                        CoT=0;
                        dirrHoT=0;
                        dirrCoT=0;
                    }
else if(counter>=5 && (strcmp("MN", getName()) == 0))
                    {
        delete ttmsg;
        counter=0;
        //lostConnection=0;

                        // Generate another one.
                        EV << "Generating another message: ";
                        TicTocMsg15 *newmsg = generateBU();
```

```
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
                        numSent++;
                }
        else if((counterA>=5) && (strcmp("Attacker", getName()) == 0)&&(strcmp(msg->getName(),"PacketReply")==0))
                {
                delete ttmsg;
                counterA=0;
                bubble("CounterA");
                packetCount=(packetCount-1);

                                // Generate another one.
                                EV << "Counter is more than 5. Generating another BU: ";
                                TicTocMsg15 *newmsg = generateBU();
                                EV << newmsg << endl;
                                forwardMessage(newmsg);
                                numSent++;
                }
        else if(strcmp(msg->getName(),"BU")==0)
                {
                delete ttmsg;

                if((dap==1)&&(rr==0))
                {
                        EV << "Starting Distributed Authentication: ";
                                                TicTocMsg15 *newmsg = requestdapMN();
                                                TicTocMsg15 *newmsg2 = requestdapHA();
                                                //EV << newmsg << endl;
                                                forwardMessage(newmsg);
                                                numSent++;
                                                forwardMessage(newmsg2);
                                                numSent++;
                }
                else if((dap==1)&&(rr==1))
                {
                        // Starting Return Routability.
                                                EV << "Starting Return Routability: ";
                                                TicTocMsg15 *newmsg = generateCoT();
                                                TicTocMsg15 *newmsg2 = generateHoT();
                                                EV << newmsg << endl;
                                                forwardMessage(newmsg);
                                                numSent++;
                                                forwardMessage(newmsg2);
                                                numSent++;

                                                if(dirr==1)
                                        {
                                                EV << "Starting dual identity Return Routability: ";
                                                TicTocMsg15 *newmsg3 = generatedirrCoT();
                                                TicTocMsg15 *newmsg4 = generatedirrHoT();
                                                EV << newmsg << endl;
                                                forwardMessage(newmsg3);
                                                numSent++;
                                                forwardMessage(newmsg4);
                                                numSent++;
                                                                        }

/*

                        EV << "Starting Distributed Authentication: ";
                                                TicTocMsg15 *newmsg3 = requestdapMN();
                                                TicTocMsg15 *newmsg4 = requestdapHA();
                                                //EV << newmsg << endl;
                                                forwardMessage(newmsg3);
                                                numSent++;
                                                forwardMessage(newmsg4);
                                                numSent++;*/

                }
        else if(rr==0)
{
                        // Generate another one.
                        EV << "Generating another message: ";
                        TicTocMsg15 *newmsg = generateBA();
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
```

```
                        numSent++;
}
else if(rr==1)
{
        // Starting Return Routability.
                        EV << "Starting Return Routability: ";
                        TicTocMsg15 *newmsg = generateCoT();
                        TicTocMsg15 *newmsg2 = generateHoT();
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
                        numSent++;
                        forwardMessage(newmsg2);
                        numSent++;
                        if(dirr==1)
                        {
                                EV << "Starting dual identity Return Routability: ";
                                                TicTocMsg15 *newmsg3 =
generatedirrCoT();
                                                TicTocMsg15 *newmsg4 =
generatedirrHoT();
                                                EV << newmsg << endl;
                                                forwardMessage(newmsg3);
                                                numSent++;
                                                forwardMessage(newmsg4);
                                                numSent++;
                        }
}
                }


    else if(strcmp(msg->getName(),"Binding-KEY")==0)
                {
                delete ttmsg;
if(dap==1)
{
                EV << "Starting Distributed Authentication: ";
                 TicTocMsg15 *newmsg3 = requestdapMN();
                TicTocMsg15 *newmsg4 = requestdapHA();
                //EV << newmsg << endl;
                 forwardMessage(newmsg3);
                numSent++;
                forwardMessage(newmsg4);
                numSent++;
}
else
{



                        // Generate another one.
                        EV << "Generating another message: ";
                        TicTocMsg15 *newmsg = generateBA();
                        EV << newmsg << endl;
                        forwardMessage(newmsg);
                        numSent++;
}

                }
    else if ((strcmp(ttmsg->getName(),"HoT")==0)&&(strcmp("AccessPoint", getName()) == 0)&&(MHA==1))
            {

                //delete ttmsg;
              bubble("Tunneling to MN");
                        // Generate another one.
                        EV << "Tunneling to MN: ";
                       // TicTocMsg15 *newmsg = tunnelMg();
                        ttmsg->setDestination(MN);
                        EV << ttmsg << endl;
                        forwardMessage(ttmsg);
                        numSent++;
            }
    else if ((strcmp(ttmsg->getName(),"HoT")==0)&&((strcmp("HA", getName()) == 0)))
        {
```

```
        //delete ttmsg;
          bubble("Tunneling to MN");
                        // Generate another one.
                        EV << "Tunneling to MN: ";
                       // TicTocMsg15 *newmsg = tunnelMg();
                        ttmsg->setDestination(MN);
                        EV << ttmsg << endl;
                        forwardMessage(ttmsg);
                        numSent++;
      }
else if ((strcmp(ttmsg->getName(),"dirrHoT")==0)&&((strcmp("AccessPoint", getName()) == 0)))
            {

                  //delete ttmsg;
               bubble("Tunneling to MN");
                                 // Generate another one.
                                 EV << "Tunneling to MN: ";
                                // TicTocMsg15 *newmsg = tunnelMg();
                                 ttmsg->setDestination(MNdirr);
                                 EV << ttmsg << endl;
                                 forwardMessage(ttmsg);
                                 numSent++;
            }
else if ((strcmp(ttmsg->getName(),"HoT")==0)&&(strcmp("Attacker", getName()) == 0))
                  {
        delete ttmsg;
         // Generate another one.
        HoT=1;
        if(((HoT==1)&&(CoT==1))&&((dirrHoT==1)&&(dirrCoT==1)))
                                          {
                                                   EV << "Sending key: ";
                                                  TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                                  //ttmsg->setDestination(MN);
                                                  EV << newmsg << endl;
                                                  forwardMessage(newmsg);
                                                  numSent++;
                                                  HoT=0;
                                                  CoT=0;
                                                  dirrHoT=0;
                                                  dirrCoT=0;
                                                  packetCount=(packetCount-1);
                                          }
                     else if((HoT==1)&&(CoT==1)&&(dirr==0))
                     {
                                 EV << "Sending key: ";
                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                //ttmsg->setDestination(MN);
                                EV << newmsg << endl;
                                forwardMessage(newmsg);
                                numSent++;
                                HoT=0;
                                CoT=0;
                                packetCount=(packetCount-1);
                     }
                      else{
                                 EV << "HoT Received: ";
                                 bubble("HoT Received");
                     }



                  }
else if (strcmp(ttmsg->getName(),"HoT")==0)
            {
        delete ttmsg;
         // Generate another one.
        HoT=1;


        if(((HoT==1)&&(CoT==1))&&((dirrHoT==1)&&(dirrCoT==1)))
                                {
                                         EV << "Sending key: ";
                                        TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                        //ttmsg->setDestination(MN);
                                        EV << newmsg << endl;
```

```
                                        forwardMessage(newmsg);
                                        numSent++;
                                        HoT=0;
                                        CoT=0;
                                        dirrHoT=0;
                                        dirrCoT=0;
                                }
        else if((HoT==1)&&(CoT==1)&&(dirr==0))
                        {
                                EV << "Sending key: ";
                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                //ttmsg->setDestination(MN);
                                EV << newmsg << endl;
                                forwardMessage(newmsg);
                                numSent++;
                                HoT=0;
                                CoT=0;
                        }
                        else{
                                EV << "HoT Received: ";
                                bubble("HoT Received");
                        }


            }
else if (strcmp(ttmsg->getName(),"dirrHoT")==0)
                {
        delete ttmsg;
        // Generate another one.
        dirrHoT=1;
        if(((HoT==1)&&(CoT==1))&&((dirrHoT==1)&&(dirrCoT==1)))
                        {
                                EV << "Sending key: ";
                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                //ttmsg->setDestination(MN);
                                EV << newmsg << endl;
                                forwardMessage(newmsg);
                                numSent++;
                                HoT=0;
                                CoT=0;
                                dirrHoT=0;
                                dirrCoT=0;
                        }
                        else{
                                EV << "dirrHoT Received: ";
                                bubble("dirrHoT Received");
                        }


                }
else if ((strcmp(msg->getName(),"CoT")==0)&&(strcmp("Attacker", getName()) == 0))
                                {
                        delete ttmsg;
                        CoT=1;
                        if(((HoT==1)&&(CoT==1))&&((dirrHoT==1)&&(dirrCoT==1)))
                                                        {
                                                                EV << "Sending key: ";
                                                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                                                //ttmsg->setDestination(MN);
                                                                EV << newmsg << endl;
                                                                forwardMessage(newmsg);
                                                                numSent++;
                                                                HoT=0;
                                                                CoT=0;
                                                                dirrHoT=0;
                                                                dirrCoT=0;
                                                                packetCount=(packetCount-1);
                                                        }
                                        else if((HoT==1)&&(CoT==1)&&(dirr==0))
                                                        {
                                                                EV << "Sending key: ";
                                                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                                                //ttmsg->setDestination(MN);
```

```
                                        EV << newmsg << endl;
                                        forwardMessage(newmsg);
                                        numSent++;
                                        HoT=0;
                                        CoT=0;
                                        packetCount=(packetCount-1);
                                }
                                else{
                                        EV << "CoT Received: ";
                                        bubble("CoT Received");
                                }

                        }
else if(strcmp(msg->getName(),"CoT")==0)
                        {
                        delete ttmsg;
                        CoT=1;

                        if(((HoT==1)&&(CoT==1))&&((dirrHoT==1)&&(dirrCoT==1)))
                                        {
                                                EV << "Sending key: ";
                                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                                //ttmsg->setDestination(MN);
                                                EV << newmsg << endl;
                                                forwardMessage(newmsg);
                                                numSent++;
                                                HoT=0;
                                                CoT=0;
                                                dirrHoT=0;
                                                dirrCoT=0;
                                        }
                        else if((HoT==1)&&(CoT==1)&&(dirr==0))
                                        {
                                                EV << "Sending key: ";
                                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                                //ttmsg->setDestination(MN);
                                                EV << newmsg << endl;
                                                forwardMessage(newmsg);
                                                numSent++;
                                                HoT=0;
                                                CoT=0;

                                        }
                                        else{
                                                EV << "CoT Received: ";
                                                bubble("CoT Received");
                                        }

                        }
else if(strcmp(msg->getName(),"dirrCoT")==0)
                        {
                        delete ttmsg;
                        dirrCoT=1;

                        if(((HoT==1)&&(CoT==1))&&((dirrHoT==1)&&(dirrCoT==1)))
                                        {
                                                EV << "Sending key: ";
                                                TicTocMsg15 *newmsg = generateKey();//tunnelMg();
                                                //ttmsg->setDestination(MN);
                                                EV << newmsg << endl;
                                                forwardMessage(newmsg);
                                                numSent++;
                                                dirrHoT=0;
                                                CoT=0;
                                                HoT=0;
                                                dirrCoT=0;

                                        }
                                        else{
                                                EV << "dirrCoT Received: ";
                                                bubble("dirrCoT Received");
                                        }

                        }
```

```
else if((strcmp(msg->getName(),"BA")==0)&&(strcmp("Attacker", getName()) == 0))
                                {
                        //delete ttmsg;




                                        EV << packetCount << endl;
                                                                delete ttmsg;
                                                                packetCount=(packetCount-1);
                                                                EV << packetCount << endl;
                                                                if(packetCount<1)
                                                                {
                                                                        // Generate another one.
                                                                EV << "Generating another message: ";
                                                                TicTocMsg15 *newmsg = generatePackets();
                                                                EV << newmsg << endl;
                                                                forwardMessage(newmsg);
                                                                EV << "Updating HA: ";
                                                                TicTocMsg15 *newmsg2 = updatingHA();
                                                                EV << newmsg2 << endl;
                                                                forwardMessage(newmsg2);
                                                                numSent++;
                                                                }
                                                                else if(packetCount>=1)
                                                                {
                                                                        bubble("message deleted");

                                                                        EV << "Deleted duplicate message: "<< endl;


                                                                }
                                }
else if(strcmp(msg->getName(),"BA")==0)
                        {
                        delete ttmsg;

                                        // Generate another one.
                                        EV << "Generating another message: ";
                                        TicTocMsg15 *newmsg = generatePackets();
                                        EV << newmsg << endl;
                                        forwardMessage(newmsg);
                                        EV << "Updating HA: ";
                                        TicTocMsg15 *newmsg2 = updatingHA();
                                        EV << newmsg2 << endl;
                                       forwardMessage(newmsg2);
                                        numSent++;
                        }

else if(strcmp(msg->getName(),"PacketRequest")==0)
                                {
                        delete ttmsg;

                                        // Generate another one.
                                        EV << "Generating another message: ";
                                        TicTocMsg15 *newmsg = generatePackets();
                                        EV << newmsg << endl;
                                        forwardMessage(newmsg);
                                        numSent++;


                                }
else if((strcmp(msg->getName(),"PacketReply")==0)&&(strcmp("Attacker", getName()) == 0))
                                {
                                EV << packetCount << endl;
                                delete ttmsg;
                                packetCount=(packetCount-1);
                                EV << packetCount << endl;
                                if(packetCount<1)
                                {
                                // Generate another one.

                                        EV << "Generating another message: ";
                                        TicTocMsg15 *newmsg = generatePackets();
```

```
                                        EV << newmsg << endl;
                                        forwardMessage(newmsg);
                                        numSent++;
                                }
                                else if(packetCount>=1)
                                {
                                        bubble("message deleted");
                                        EV << "Deleted duplicate message: "<< endl;



                                }
                                }
        else if(strcmp(msg->getName(),"PacketReply")==0)
                                {

                                delete ttmsg;

                                        EV << "Generating another message: ";
                                        TicTocMsg15 *newmsg = generatePackets();
                                        EV << newmsg << endl;
                                        forwardMessage(newmsg);
                                        numSent++;


                                }
        else if(strcmp(msg->getName(),"start")==0)
                                {
                                delete ttmsg;

                                        // Generate another one.
                                        EV << "Starting Attack: ";
                                        TicTocMsg15 *newmsg = attackST();
                                        EV << newmsg << endl;
                                        forwardMessage(newmsg);
                                        numSent++;
                                }
      else
       {
              delete ttmsg;
              bubble("else gen mesg");
                    // Generate another one.
                    EV << "Generating another message: ";
                    TicTocMsg15 *newmsg = generateMessage();
                    EV << newmsg << endl;
                    forwardMessage(newmsg);
                    numSent++;
       }



    }
    else
    {
      // We need to forward the message.
      forwardMessage(ttmsg);
    }

       }//event
       }

TicTocMsg15 *Node::generateMessage()
{
              //int src;
              //int dest;
              char msgname[20];
              //int CoA;
              // Produce source and destination addresses.
              if (strcmp("MN", getName()) == 0)
              {

if(cgasecurity==0)
{
              if(MHA==0)
              {
```

```
		src = HA;//getIndex();
		dest = CN ;
		}
		else if (MHA==1)
		{
			src = accesspoint;//getIndex();
			dest = CN ;
		}
		//CA = 1;
		//int n = size();
		sprintf(msgname, "Request");
}
else if(cgasecurity==1)
{
		if(MHA==0)
		{
		src = HA;//getIndex();
		dest = CN ;
		}
		else if(MHA==1)
		{
			src = accesspoint;//getIndex();
			dest = CN ;
		}
			//CA = 1;
			//int n = size();
			sprintf(msgname, "CGA-Request");
}

		}
		else if (strcmp("CN", getName()) == 0)
			{
			CNHABuffer=packetsource;

			src = CN;//getIndex();
		  //int n = size();
			dest = packetsource;
			//sprintf(msgname, "Acknowledgement-%d-to-%d", src, dest);
			sprintf(msgname, "Acknowledgement");
			}
		else if (strcmp("Attacker", getName()) == 0)
				{

			if (attackercga==0)
			{

			src = attacker;//getIndex();
			  //int n = size();
				dest = CN;
				//sprintf(msgname, "Acknowledgement-%d-to-%d", src, dest);
				sprintf(msgname, "Request");
			}
			else if(attackercga==1)
			{
				src = attacker;//getIndex();
					//int n = size();
					dest = CN;
					sprintf(msgname, "CGA-Request");
			}
			else if(attackercga==2)
				{
				src = attacker;//getIndex();
					//int n = size();
					dest = CN;
					sprintf(msgname, "CGA-Request");
				}


			}


		//if (strcmp("MN", getName()) == 0)
		//int dest = ;//intuniform(0,n-2);
```

```
    //if (dest>=src) dest++;


    //sprintf(msgname, "message-%d-to-%d", src, dest);

    // Create message object and set source and destination field.
    TicTocMsg15 *msg = new TicTocMsg15(msgname);
    msg->setSource(src);
    msg->setDestination(dest);

    //msg->setCoA(CN);
    return msg;
}

TicTocMsg15 *Node::dapMNauth()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                if (strcmp("MN", getName()) == 0)
                {
                src = MN;//getIndex();
                dest = packetsource;
                //CA = 1;
                //int n = size();
                sprintf(msgname, "dapMNauth");
                }
                if (strcmp("Attacker", getName()) == 0)
                                {
                                src = attacker;//getIndex();
                                dest = packetsource;
                                //CA = 1;
                                //int n = size();
                                sprintf(msgname, "dapMNauth");
                                //packetCount++;
                                }
    // Create message object and set source and destination field.
    TicTocMsg15 *msg = new TicTocMsg15(msgname);
    msg->setSource(src);
    msg->setDestination(dest);
    //msg->setCoA(CN);
    return msg;
}
TicTocMsg15 *Node::dapHAauth()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                if (strcmp("HA", getName()) == 0)
                {
                src = HA;//getIndex();
                dest = CN;//packetsource;
                //CA = 1;
                //int n = size();
                sprintf(msgname, "dapHAauth");
                }
                /*if (strcmp("Attacker", getName()) == 0)
                                {
                                src = attacker;//getIndex();
                                dest = packetsource;
                                //CA = 1;
                                //int n = size();
                                sprintf(msgname, "BU");
                                packetCount++;
                                }*/
    // Create message object and set source and destination field.
    TicTocMsg15 *msg = new TicTocMsg15(msgname);
    msg->setSource(src);
    msg->setDestination(dest);
```

```
   //msg->setCoA(CN);
   return msg;
}
TicTocMsg15 *Node::generateBU()
{
		//int src;
		//int dest;
		char msgname[20];
		//int CoA;
		// Produce source and destination addresses.
		if (strcmp("MN", getName()) == 0)
		{
		src = MN;//getIndex();
		dest = packetsource;
		//CA = 1;
		//int n = size();
		sprintf(msgname, "BU");
		}
		if (strcmp("Attacker", getName()) == 0)
			{
			src = attacker;//getIndex();
			dest = packetsource;
			//CA = 1;
			//int n = size();
			sprintf(msgname, "BU");
			packetCount++;
			}
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}
TicTocMsg15 *Node::generateKey()
{
		//int src;
		//int dest;
		char msgname[20];
		//int CoA;
		// Produce source and destination addresses.
		if (strcmp("MN", getName()) == 0)
		{
		src = MN;//getIndex();
		dest = packetsource;
		//CA = 1;
		//int n = size();
		sprintf(msgname, "Binding-KEY");
		}
		if (strcmp("Attacker", getName()) == 0)
			{
			src = attacker;//getIndex();
			dest = packetsource;
			//CA = 1;
			//int n = size();
			sprintf(msgname, "Binding-KEY");
			packetCount++;
			}
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}



TicTocMsg15 *Node::tunnelMg()
{
		//int src;
			//int dest;
			char msgname[20];
			//int CoA;
			// Produce source and destination addresses.
```

```
                        //if (strcmp("MN", getName()) == 0)
                        //{
                        src = MN;//getIndex();
                        dest = CN;
                        //CA = 1;
                        //int n = size();
                        sprintf(msgname, "BU");
                        //}
                // Create message object and set source and destination field.
                TicTocMsg15 *msg = new TicTocMsg15(msgname);
                msg->setSource(src);
                msg->setDestination(dest);
                //msg->setCoA(CN);
                return msg;
}
TicTocMsg15 *Node::attackMsg()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                //if (strcmp("CN", getName()) == 0)
                //{
                src = CN;//getIndex();
                dest = attacker;
                //CA = 1;
                //int n = size();
                sprintf(msgname, "start");
                //}
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}
TicTocMsg15 *Node::attackST()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                //if (strcmp("CN", getName()) == 0)
                //{
                src = attacker;//getIndex();
                dest = CN;
                //CA = 1;
                //int n = size();
                sprintf(msgname, "BU");
                //}
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}
TicTocMsg15 *Node::generateBA()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                if (strcmp("CN", getName()) == 0)
                {
                src = CN;//getIndex();
                dest = packetsource;
                //CA = 1;
                //int n = size();
                sprintf(msgname, "BA");
                }
   // Create message object and set source and destination field.
```

```
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}


TicTocMsg15 *Node::requestdapMN()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                if (strcmp("CN", getName()) == 0)
                {
                src = CN;//getIndex();
                dest = packetsource;
                //CA = 1;
                //int n = size();
                sprintf(msgname, "requestdapMN");
                }
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}


TicTocMsg15 *Node::requestdapHA()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                if (strcmp("CN", getName()) == 0)
                {

                if(MHA==0)
                {
                src = CN;//getIndex();
                dest = HA;
                }
                else if(MHA==1)
                {
                                src = CN;//getIndex();
                                dest = accesspoint;
                }
                //CA = 1;
                //int n = size();
                sprintf(msgname, "requestdapHA");
                }
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}

TicTocMsg15 *Node::generateHoT()
{
                //int src;
                //int dest;
                char msgname[20];
                //int CoA;
                // Produce source and destination addresses.
                if (strcmp("CN", getName()) == 0)
                {

                if(MHA==0)
```

```
        {
                if(rrAttack==1)
                        {
                        src = CN;//getIndex();
                        dest = HA;
                        //CA = 1;
                        //int n = size();
                        sprintf(msgname, "HoT");
                        }
        else if(rrAttack==2)
        {
                if(packetsource==attacker)
                                                {
                                                src = CN;//getIndex();
                                                dest = packetsource;
                                                }
                                                else if(packetsource==MN)
                                                {
                                                        src = CN;//getIndex();
                                                        dest = HA;
                                                }

                //src = CN;//getIndex();
        //dest = CNHABuffer;
        //CA = 1;
        //int n = size();
        sprintf(msgname, "HoT");
        }
        }
        else if(MHA==1)
        {
                /*src = CN;//getIndex();
                dest = accesspoint;
                sprintf(msgname, "HoT");*/
                if(rrAttack==1)
                                                {
                                                src = CN;//getIndex();
                                                dest = accesspoint;//HA;
                                                //CA = 1;
                                                //int n = size();
                                                sprintf(msgname, "HoT");
                                                }
                        else if(rrAttack==2)
                        {
                        if(packetsource==attacker)
                        {
                        src = CN;//getIndex();
                        dest = packetsource;
                        }
                        else if(packetsource==MN)
                        {
                                src = CN;//getIndex();
                                dest = accesspoint;
                        }
                        //CA = 1;
                        //int n = size();
                        sprintf(msgname, "HoT");
                        }



        }
        }
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}
TicTocMsg15 *Node::generatedirrHoT()
{
        //int src;
        //int dest;
        char msgname[20];
```

```
		//int CoA;
		// Produce source and destination addresses.
		if (strcmp("CN", getName()) == 0)
		{

			//if(dirrAttack==1)
			//	{

				src = CN;//getIndex();
				dest = accesspoint;


				//CA = 1;
				//int n = size();
				sprintf(msgname, "dirrHoT");
			//	}
		/*else if(rrAttack==2)
		{
		src = CN;//getIndex();
		dest = CNHABuffer;
		//CA = 1;
		//int n = size();
		sprintf(msgname, "HoT");
		}*/

		}
	// Create message object and set source and destination field.
	TicTocMsg15 *msg = new TicTocMsg15(msgname);
	msg->setSource(src);
	msg->setDestination(dest);
	//msg->setCoA(CN);
	return msg;
}
TicTocMsg15 *Node::generateCoT()
{
		//int src;
		//int dest;
		char msgname[20];
		//int CoA;
		// Produce source and destination addresses.
		if (strcmp("CN", getName()) == 0)
		{
		src = CN;//getIndex();
		dest = packetsource;
		//CA = 1;
		//int n = size();
		sprintf(msgname, "CoT");
		}
	// Create message object and set source and destination field.
	TicTocMsg15 *msg = new TicTocMsg15(msgname);
	msg->setSource(src);
	msg->setDestination(dest);
	//msg->setCoA(CN);
	return msg;
}
TicTocMsg15 *Node::generatedirrCoT()
{
		//int src;
		//int dest;
		char msgname[20];
		//int CoA;
		// Produce source and destination addresses.
		if (strcmp("CN", getName()) == 0)
		{
		src = CN;//getIndex();
		dest = MNdirr;
		//CA = 1;
		//int n = size();
		sprintf(msgname, "dirrCoT");
		}
	// Create message object and set source and destination field.
	TicTocMsg15 *msg = new TicTocMsg15(msgname);
	msg->setSource(src);
	msg->setDestination(dest);
	//msg->setCoA(CN);
```

```
   return msg;
}

TicTocMsg15 *Node::updatingHA()
{
		//int src;
		//int dest;
		char msgname[20];
		//int CoA;
		// Produce source and destination addresses.
		if (strcmp("MN", getName()) == 0)
		{
		if (MHA==0)
		{
		src = MN;//getIndex();
		dest = HA;
		}
		else if (MHA==1)
		{
				src = MN;//getIndex();
				dest = accesspoint;
		}
		//CA = 1;
		//int n = size();
		sprintf(msgname, "CoA");
		}
		else if (strcmp("Attacker", getName()) == 0)
				{
				src = attacker;//getIndex();
				dest = HA;
				//CA = 1;
				//int n = size();
				sprintf(msgname, "CoA");
				}
   // Create message object and set source and destination field.
   TicTocMsg15 *msg = new TicTocMsg15(msgname);
   msg->setSource(src);
   msg->setDestination(dest);
   //msg->setCoA(CN);
   return msg;
}
TicTocMsg15 *Node::generatePackets()
{
		//int src;
		//int dest;
		char msgname[20];
		counter++;
		//counterA++;
		//int CoA;
		// Produce source and destination addresses.
		if (strcmp("MN", getName()) == 0)
		{
		src = MN;//getIndex();
		dest = packetsource;
		//CA = 1;
		//int n = size();
		sprintf(msgname, "PacketRequest");
		}
		else if (strcmp("CN", getName()) == 0)
				{
				src = CN;//getIndex();
				dest = packetsource;
				//CA = 1;
				//int n = size();
				sprintf(msgname, "PacketReply");
				}
		else if (strcmp("Attacker", getName()) == 0)
					{
					src = attacker;//getIndex();
					dest = packetsource;
					counterA++;
					//CA = 1;
					//int n = size();
					sprintf(msgname, "PacketRequest");
					packetCount++;
```

```
                                        }
    // Create message object and set source and destination field.
    TicTocMsg15 *msg = new TicTocMsg15(msgname);
    msg->setSource(src);
    msg->setDestination(dest);
    //msg->setCoA(CN);
    return msg;
}



void Node::forwardMessage(TicTocMsg15 *msg)
{
    // Increment hop count.
    msg->setHopCount(msg->getHopCount()+1);

    // Same routing as before: random gate.
    //int n = gateSize("gate");
    int k;// = 0;//intuniform(0,n-1);
    if (strcmp("MN", getName()) == 0)
                        {
                if (msg->getDestination()==attacker)
                {
                        k=1;
                }
                else if(home==1)
                {
                        k=0;
                }
                else if(home==0)
                        {
                                k=2;
                        }

                        }
    if (strcmp("PoA", getName()) == 0)
                        {
                GateID=msg->getArrivalGate()->getIndex();

                if (msg->getDestination()==attacker)
                        {
                                k=2;
                        }
                else if ((msg->getDestination()==MN)&&(home==1))
                //if (GateID == 0)
                                {
                                k=1;
                                }
                else if ((msg->getDestination()==MN)&&(home==0))
                        //if (GateID == 0)
                                                {
                                                k=0;
                                                }
                                        //if (GateID == 1)

                else
                {
                 k=0;
                }
                //k=0;
                        }
    if (strcmp("HA", getName()) == 0)
                        {
                GateID=msg->getArrivalGate()->getIndex();
                        //if (GateID == 0)
                if ((msg->getDestination()==MN)&&(home==1))
                                                {
                        k=1;
                        bubble("k1 MN Home 1");
                                                }
                else if ((msg->getDestination()==MN)&&(home==0))
                                                        {
                                k=0;
                                bubble("k0 MN home 0");
```

```
                                        }
                else if ((msg->getDestination()==attacker)&&(directAttack==1))
                                                                //if (GateID == 1)
                                {
                                k=2;
                                bubble("Attacker direct k=2");
                                }
                else if ((msg->getDestination()==attacker)&&(directAttack==0))
                                                                        //if (GateID == 1)
                                        {
                                        k=1;
                                        bubble("k1 attacker direct 0");
                                        }
                else if (msg->getDestination()==CN)
                                                                        //if (GateID == 1)
                                        {
                                        k=0;
                                        bubble("k0 CN");
                                        }
                //else if (msg->getDestination()==HA)
                //{
                //k=0;
                 //             bubble("k0 HA");
                //}

                else
                                        {
                                        k=0;
                                        bubble("else k0");
                                        }
                //k=0;
                                }

if (strcmp("Router1", getName()) == 0)
                                {
                GateID=msg->getArrivalGate()->getIndex();
                //if (GateID == 1)
                //bubble(packetsource);
                if ((msg->getDestination()==MN)&&(home==1))
                {
                k=1;
                bubble("k1 MN home1");
                }
                else if ((msg->getDestination()==MN)&&(home==0))
                                {
                                k=0;
                                bubble("k0");
                                }
                else if (msg->getDestination()==MNdirr)
                                        {
                                        k=0;
                                        bubble("k0");
                                        }
                else if (msg->getDestination()==accesspoint)
                                                                {
                                                                k=0;
                                                                bubble("k0");
                                                                }
                //if (GateID == 3)
                //else if (msg->getDestination()==5)
                else if ((msg->getDestination()==CN)&&(mobile==0))
                                {
                                k=3;
                                bubble("k3");
                                }
                else if ((msg->getDestination()==CN)&&(mobile==1))
                                        {
                                        k=4;
                                        bubble("k4");
                                        }
                else if ((msg->getDestination()==attacker)&&(directAttack==1))
                                        {
                                        k=2;
                                        bubble("k2");
                                        }
```

```
        else if ((msg->getDestination()==attacker)&&(directAttack==0))
                                        {
                                        k=1;
                                        bubble("k1");
                                        }
        else if((msg->getDestination()==HA)&&(mobile==0))
        {
                k=1;
                        bubble("HA routing");
        }
        //else if (msg->getDestination()==11)
                        //              {
                        //              k=5;
                        //              }
        else
                                                {
                                                k=1;
                                                bubble("else k1");
                                                }

                }


if (strcmp("CN", getName()) == 0)
                {
        if ((msg->getDestination()==MN)&&(mobile==0))
                {
                k=1;
                }
        else if ((msg->getDestination()==MNdirr)&&(mobile==0))
                                {
                                k=1;
                                }
        else if ((msg->getDestination()==accesspoint)&&(mobile==0))
                                        {
                                        k=1;
                                        }
        else if ((msg->getDestination()==attacker)&&(directAttack==1))
                                {
                                k=0;
                                }
        else if ((msg->getDestination()==attacker)&&(directAttack==0))
                                        {
                                        k=1;
                                        }
        else if ((msg->getDestination()==HA)&&(mobile==0))
                                        {
                                        k=1;
                                        }
        else
        {
        k=2;
                }
                }
if (strcmp("Attacker", getName()) == 0)
                {
if(directAttack==1)
{
        if (msg->getDestination()==CN)
                {
                k=4;
                bubble("k=4");
                }
        else if (msg->getDestination()==MN)
                                {
                                k=0;
                                bubble("k=0");
                                }
        else if (msg->getDestination()==PoA)
                                {
                                k=1;
                                bubble("k=1");
                                }
        else if (msg->getDestination()==HA)
                                        {
```

```
                                        k=2;
                                        bubble("k=2");
                                        }
        else if (msg->getDestination()==Router1)
                                        {
                                        k=3;
                                        bubble("k=3");
                                        }
        else if (msg->getDestination()==CHA)
                                        {
                                        k=7;
                                        bubble("k=7");
                                        }
        else if (msg->getDestination()==Router2)
                                        {
                                        k=5;
                                        bubble("k=5");
                                        }
        else if (msg->getDestination()==accesspoint)
                                        {
                                        k=6;
                                        bubble("k=1");
                                        }
        else
        {
        k=0;
        bubble("else k=0");
        }
}
else if (directAttack==0)
{
        if (msg->getDestination()==CN)
                                {
                                k=1;
                                bubble("k=1");
                                }
                else if (msg->getDestination()==MN)
                                        {
                                        k=0;
                                        bubble("k=0");
                                        }
                else if (msg->getDestination()==PoA)
                                        {
                                        k=1;
                                        bubble("k=1");
                                        }
                else if (msg->getDestination()==HA)
                                                {
                                                k=1;
                                                bubble("k=1");
                                                }
                else if (msg->getDestination()==Router1)
                                                {
                                                k=3;
                                                bubble("k=3");
                                                }
                else if (msg->getDestination()==CHA)
                                                {
                                                k=7;
                                                bubble("k=7");
                                                }
                else if (msg->getDestination()==Router2)
                                                {
                                                k=5;
                                                bubble("k=5");
                                                }
                else if (msg->getDestination()==accesspoint)
                                                {
                                                k=6;
                                                bubble("k=1");
                                                }
                else
                {
                k=0;
                bubble("else k=0");
```

```
                }
}
                }

if (strcmp("CHA", getName()) == 0)
                {
        GateID=msg->getArrivalGate()->getIndex();
                //if (GateID == 0)
        if ((msg->getDestination()==MN)&&(home==1))
                                {
                k=1;
                bubble("k1 h=1");
                                }
        else if ((msg->getDestination()==MN)&&(home==0))
                                        {
                        k=1;
                        bubble("k1 h=0");
                                        }
        else if (msg->getDestination()==MNdirr)
                                {
                                k=1;
                                bubble("k1");
                                }
                else if (msg->getDestination()==accesspoint)
                                                {
                                                k=1;
                                                bubble("k1");
                                                }
        else if ((msg->getDestination()==HA)&&(home==1))
                                        {
                        k=1;
                        bubble("HA k1 h=1");
                                        }
                else if ((msg->getDestination()==HA)&&(home==0))
                                                {
                                k=1;
                                bubble("HA k1 h=0");
                                                }
        else if (msg->getDestination()==attacker)
                                                //if (GateID == 1)
                {
                k=0;
                bubble("k0 attacker");
                }
        else
                        {
                        k=2;
                        bubble("k2 else");
                        }
        //k=0;
                }
if (strcmp("Router2", getName()) == 0)
                {
        GateID=msg->getArrivalGate()->getIndex();
                //if (GateID == 0)
        if ((msg->getDestination()==MN)&&(home==0))
                                {
                k=1;
                }
        else if ((msg->getDestination()==MN)&&(home==1))
                                        {
                        k=0;
                        }
        else if (msg->getDestination()==MNdirr)
                                {
                                k=1;
                                bubble("k1");
                                }
        else if (msg->getDestination()==accesspoint)
                                                {
                                                k=1;
                                                bubble("k1");
                                                }
        else if (msg->getDestination()==attacker)
                                                //if (GateID == 1)
```

```
                {
                k=2;
                }
        else
                        {
                        k=0;
                        }
        //k=0;
                }
  if (strcmp("AccessPoint", getName()) == 0)
                {
        GateID=msg->getArrivalGate()->getIndex();
                //if (GateID == 0)
        if ((msg->getDestination()==MN)&&(home==0))
                                {
                k=2;
                bubble("k2");
                                }
        else if ((msg->getDestination()==MN)&&(home==1))
                                        {
                        k=0;
                        bubble("k0");
                                        }
        else if (msg->getDestination()==MNdirr)
                                {
                                k=2;
                                bubble("k2");
                                }
        /*else if (msg->getDestination()==accesspoint)
                                        {
                                        k=2;
                                        bubble("k2");
                                        }*/
        else if (msg->getDestination()==attacker)
                                        //if (GateID == 1)
                {
                k=1;
                bubble("k1");
                }
        else
                        {
                        k=0;
                        bubble("else k0");
                        }
        //k=0;
                }

  EV << "Forwarding message " << msg << " on gate[" << k << "]\n";
  send(msg, "gate$o", k);
}

void Node::finish()
{
  // This function is called by OMNeT++ at the end of the simulation.
        EV << "Module:     " << getName() << endl;
        EV << "Sent:     " << numSent << endl;
  EV << "Received: " << numReceived << endl;
  EV << "Hop count, min:    " << hopCountStats.getMin() << endl;
  EV << "Hop count, max:    " << hopCountStats.getMax() << endl;
  EV << "Hop count, mean:   " << hopCountStats.getMean() << endl;
  EV << "Hop count, stddev: " << hopCountStats.getStddev() << endl << endl;

  recordScalar("#sent", numSent);
  recordScalar("#received", numReceived);

  hopCountStats.recordAs("hop count");
}
```